



Universidad
Complutense
Madrid



Facultad
de
Informática

PROYECTO DE SISTEMAS INFORMÁTICOS

CURSO 2012 - 2013

HERRAMIENTA DE APOYO A LA ENSEÑANZA DE LAS MATEMÁTICAS EN EDUCACIÓN PRIMARIA

Francisco Javier Heras Martín

Silvia Panadero Hernández

Estefanía Sánchez García-Velasco

Dirigido por Fernando Rubio Díez

Los abajo firmantes, matriculados en Ingeniería en Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el presente proyecto de Sistemas Informáticos: “Herramienta de apoyo a la enseñanza de las matemáticas en educación primaria”, realizado durante el curso 2012-2013 bajo la dirección de Fernando Rubio Díez en el Departamento de Sistemas Informáticos y Computación (SIC), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Fdo. Francisco Javier Heras
Martín

Fdo. Silvia Panadero
Hernández

Fdo. Estefanía Sánchez
García-Velasco

*Este trabajo se lo dedicamos a nuestras familias,
que nos han apoyado en estos años de carrera.*

Gracias.

Agradecimientos

Cuando empezamos esta carrera, veíamos muy lejos el final de la misma. Ahora que éste se aproxima, echamos la vista atrás y vemos con orgullo el esfuerzo y las horas de trabajo invertidos. Este proyecto supone el punto final de una etapa de nuestras vidas en la que hemos aprendido y trabajado pero también hemos disfrutado de nuestra experiencia universitaria. Todo ello nos ha ayudado a madurar tanto intelectual como personalmente.

Queremos agradecer a nuestras familias, parejas y amigos todo el apoyo recibido, lo que ha supuesto una gran motivación para no rendirnos en los momentos más duros. Sin ellos no habríamos llegado hasta aquí.

Hacemos extensible la dedicatoria a nuestros compañeros, por los buenos momentos.

También queremos agradecer a nuestro director de proyecto, Fernando Rubio, la oportunidad que nos ha ofrecido de desarrollar una aplicación con posible difusión en el panorama actual, por su orientación a lo largo del presente trabajo y por confiar en nosotros en todo momento.

¡Muchas gracias a todos!

Índice

Resumen	11
Abstract	12
Capítulo 1: Introducción	13
1.1. Motivación	13
1.2. Objetivos e hipótesis de trabajo	14
1.3. Solución planteada	16
1.4. Estados de arte	17
1.5. Organización del documento	19
Capítulo 2: Técnicas y herramientas utilizadas	21
2.1. Metodologías	21
2.2. Programación	22
2.3. Sistemas de almacenamiento	24
2.4. Herramientas de desarrollo	25
2.5. Herramientas de documentación	27
2.6. Herramientas de diseño gráfico	27
Capítulo 3: Estudio de viabilidad del software	28
3.1. Introducción	28
3.2. Planificación temporal del proyecto	28
3.3. Evaluación del riesgo técnico	30
Capítulo 4: Aspectos relevantes del desarrollo	31
4.1. Ciclo de vida	31
4.2. Análisis	32
4.3. Entorno tecnológico	33
4.4. Implementación	34
4.5. Documentación	35
4.6. Pruebas	35
Capítulo 5: Especificación de requisitos del software	36
5.1. Objetivos del proyecto	36
5.2. Catálogo de requisitos	41

Capítulo 6: Especificación de diseño del software	59
6.1. Introducción	59
6.2. Ámbito del software	60
6.3. Diseño	60
Capítulo 7: Técnicas de programación	69
7.1. Introducción	69
7.2. Lenguajes utilizados	70
7.3. Implementación	71
Capítulo 8: Funcionalidades de la aplicación	74
8.1. Funcionalidad para profesores	74
8.2. Funcionalidad para alumnos	84
8.3. Funcionalidad para administrador	93
Capítulo 9: Conclusiones y líneas de trabajo futuras	96
9.1. Conclusiones	96
9.2. Líneas de trabajo futuras	97
Bibliografía	98

Índice de ilustraciones

Ilustración 1: Proceso Unificado	21
Ilustración 2: PHP	22
Ilustración 3: JavaScript	22
Ilustración 4: CSS	23
Ilustración 5: HTML	23
Ilustración 6: Ajax	23
Ilustración 7: jQuery	24
Ilustración 8: MySQL	24
Ilustración 9: Apache	24
Ilustración 10: REM	25
Ilustración 11: Visual Paradigm for UML	25
Ilustración 12: WampServer	25
Ilustración 13: phpMyAdmin	26
Ilustración 14: Dreamweaver	26
Ilustración 15: MySQL Workbench	26
Ilustración 16: Gantt Project	26
Ilustración 17: Microsoft Word	27
Ilustración 18: Adobe Photoshop	27
Ilustración 19: Fechas previstas para el proyecto	29
Ilustración 20: Diagrama de Gantt	29
Ilustración 21: Diagrama de casos de uso. Gestión de alumnos.	45
Ilustración 22: Diagrama de casos de uso. Gestión de profesores.	46
Ilustración 23: Diagrama de casos de uso. Gestión de problemas.	46
Ilustración 24: Diagrama de casos de uso. Gestión de calificaciones.	47
Ilustración 25: Diagrama de casos de uso. Gestión de cursos.	47
Ilustración 26: Diagrama de casos de uso. Gestión de temas.	47
Ilustración 27: Diagrama de casos de uso. Gestión de foro.	48
Ilustración 28: Diagrama de casos de uso. Gestión de mensajes	48
Ilustración 29: Estructura de la base de datos	61
Ilustración 30: Ejemplo de interfaz antes de autenticarse los profesores	65
Ilustración 31: Ejemplo de interfaz después de autenticarse los profesores	66
Ilustración 32: Ejemplo de interfaz antes de autenticarse los alumnos	67

Ilustración 33: Ejemplo de interfaz después de autenticarse los alumnos	67
Ilustración 34: Ejemplo de interfaz administrador	68
Ilustración 35: Tabla dinámica jQuery	73
Ilustración 36: Funcionalidades profesores. Página de inicio.	74
Ilustración 37: Funcionalidades profesores. Recordatorio contraseña.	75
Ilustración 38: Funcionalidades profesores. Acceso profesores.	76
Ilustración 39: Funcionalidades profesores. Inicio registrado.	76
Ilustración 40: Funcionalidades profesores. Cursos.	77
Ilustración 41: Funcionalidades profesores. Problemas (1).	78
Ilustración 42: Funcionalidades profesores. Problemas (2).	79
Ilustración 43: Funcionalidades profesores. Crear problema.	79
Ilustración 44: Funcionalidades profesores. Problemas (3).	80
Ilustración 45: Funcionalidades profesores. Alumnos.	81
Ilustración 46: Funcionalidades profesores. Calificación de ejercicios.	82
Ilustración 47: Funcionalidades alumnos. Acceso.	84
Ilustración 48: Funcionalidades alumnos. Principal.	85
Ilustración 49: Funcionalidades alumnos. ¿Quién soy?.	85
Ilustración 50: Funcionalidades alumnos. Datos personales.	86
Ilustración 51: Funcionalidades alumnos. Problemas.	86
Ilustración 52: Funcionalidades alumnos. Problemas disponibles.	87
Ilustración 53: Funcionalidades alumnos. Resolución de problemas.	88
Ilustración 54: Funcionalidades alumnos. Calculadora.	89
Ilustración 55: Funcionalidades alumnos. Calculadora números romanos.	89
Ilustración 56: Funcionalidades alumnos. Calculadora fracciones.	90
Ilustración 57: Funcionalidades alumnos. Evaluación.	90
Ilustración 58: Funcionalidades alumnos. Calificaciones.	91
Ilustración 59: Funcionalidades alumnos. Foro.	91
Ilustración 60: Funcionalidades alumnos. Temas foro.	92
Ilustración 61: Funcionalidades alumnos. Formulario foro.	92
Ilustración 62: Funcionalidades alumnos. Respuestas foro.	93
Ilustración 63: Funcionalidades administrador. Dar de alta / Actualizar profesor.	94
Ilustración 64: Funcionalidades administrador. Dar de baja profesor.	95
Ilustración 65: Funcionalidades administrador. Enviar email.	95
Ilustración 66: Funcionalidades administrador. Listado de profesores.	95

Índice de tablas

Tabla 1: Diccionario de datos. Tabla alumnos.	62
Tabla 2: Diccionario de datos.	62
Tabla alumno_realiza_problema.		
Tabla 3: Diccionario de datos. Tabla curso.	62
Tabla 4: Diccionario de datos. Tabla problema.	63
Tabla 5: Diccionario de datos. Tabla profesor.	63
Tabla 6: Diccionario de datos. Tabla tema.	63
Tabla 7: Diccionario de datos. Tabla curso_tiene_problema.	64
Tabla 8: Diccionario de datos. Tabla foro.	64

Resumen

Herramienta de apoyo a la enseñanza de las matemáticas en educación primaria

En la actualidad, entre las materias con más dificultad para los alumnos están las matemáticas. Este tipo de materias exige una metodología y un razonamiento pragmático, por esta razón se requiere mucha concentración y ganas de aprender por parte del alumno. Un método muy comúnmente empleado es una enseñanza formal partiendo de una explicación teórica que dificulta al alumno el mantener por mucho tiempo la atención en una explicación de un contenido.

Existen muchas herramientas y técnicas de enseñanza que permiten facilitar el trabajo a los profesores. Algunas de las más extendidas son: problemas y juegos matemáticos, matemática recreativa, cálculo mental, figuras geométricas... La mayoría de ellas, realizadas en clases con un gran número de alumnos, no dan el resultado esperado debido a que el profesor no puede centrar su atención en los pasos dados por el niño durante la realización de los ejercicios.

En el presente trabajo se diseña y desarrolla un software educativo para niños de 6 a 12 años que fomenta el aprendizaje de matemáticas de una forma entretenida.

El principal objetivo de esta aplicación Web es facilitar a los alumnos y profesores de la asignatura de matemáticas, compartir y acceder a material educativo que les ayude en sus estudios.

A diferencia de otros recursos similares, esta aplicación tiene la ventaja de que son los propios profesores registrados quienes, mediante sus propias subidas de material se encargan de actualizar y controlar los contenidos para cada alumno.

A los alumnos se les plantearán los problemas que hayan definido los profesores y se les facilitarán herramientas de apoyo. La herramienta básica de la aplicación es una calculadora, donde dependiendo del problema planteado se inhabilitarán algunas teclas para forzar que el problema se resuelva mediante determinadas técnicas.

Una parte fundamental es el registro de todos los pasos que el alumno haya dado, de modo que el profesor pueda posteriormente ver una recreación de lo que ha hecho cada uno y analizar cómo ha resuelto cada problema cada alumno, así como obtener diversos datos estadísticos sobre la evolución de los alumnos.

Palabras clave: Aplicación Web, enseñanza, matemáticas, alumno, profesor, problema, calculadora.

Abstract

Learning tool for Mathematics in primary education

Nowadays, Mathematics are among the most difficult subjects for the pupils. This kind of subjects demands a methodology and pragmatic reasoning. For this reason pupils need so much concentration and a wish of learning. One of the most used method is the use of a formal education: it starts from a theoretical explanation that doesn't allow teachers hold pupils' attention for much time.

There are a lot of tools and education's technologies which make teachers work easier. Some of the most used tools are: problems and mathematical games, recreative Mathematics, mental calculation, geometric figures ... Unfortunately, when using most of them in classes with a big number of pupils, they don't provide the expected result due to the fact that the teacher cannot concentrate his attention on the steps given by the children during the accomplishment of their exercises.

In the present work it is designed and developed an educational software for children ranging 6 to 12 years that encourages mathematical learning by using an entertaining method.

The main purpose of this web application is to provide students and teachers of mathematics subject, share and access to educational materials to help them in their studies.

Unlike other similar resources, this application has the advantage that registered teachers are their own who, through their own material upload, are responsible for updating and controlling every student content.

Problems, which are defined by teachers, will appear to pupils and supporting tools will be facilitated to them. The application basic tool is a calculator. And it depends on the problem, the teacher will disqualify some keys to force pupils to solve the problem by means of certain technologies.

A fundamental part is the record of all the steps that the student has given, so that the teacher could see later a recreation of what has made each one and to analyze how every student has solved every problem, obtaining diverse statistical information on the evolution of the pupils as well.

Keywords: Web application, education, mathematics, student, teacher, problem, calculator.

Capítulo 1

Introducción

En el presente capítulo comentaremos la motivación y objetivos del proyecto, así como también la forma en la que se ha desarrollado el mismo.

1.1 MOTIVACIÓN

Hoy en día Internet y la Web han influido enormemente tanto en el mundo de la informática como en la sociedad en general. La aparición de Internet y su gran desarrollo en los últimos 10 años nos ha descubierto todo un abanico de nuevas posibilidades, entre las que se destacan la aparición de las redes sociales que ofrece la posibilidad de mantenerse en contacto con familiares y amigos a cualquier hora y en cualquier lugar, la búsqueda y acceso a información de diversos temas o la facilidad de aprendizaje y desarrollo de diferentes disciplinas. Además, una de las áreas que más expansión está teniendo en los últimos años son las aplicaciones web.

Centrándonos en el mundo de la educación, actualmente la mayoría de los colegios disponen de un aula de informática donde los alumnos pueden realizar diferentes actividades académicas en horario escolar. Además cada vez es más frecuente disponer de un terminal portátil para cada alumno, gracias al programa Escuela 2.0 del Gobierno de España, que consiste en facilitar de forma gratuita y escalonada al alumnado de 5º y 6º de Educación Primaria y 1º y 2º de ESO un ordenador portátil para trabajar en clase a través de una conexión con el profesor dentro del aula. La finalidad de este programa es poner en marcha las aulas digitales del siglo XXI, dotando de recursos técnicos a los alumnos y a los centros para garantizar la conectividad a Internet y la interconectividad dentro del aula, entre otros.

Así, surge una nueva oportunidad en el mercado de desarrollo de aplicaciones, el desarrollo para la enseñanza.

Es en este panorama donde surge la idea de una herramienta de apoyo a la enseñanza de matemáticas en educación primaria, ya que es una de las asignaturas que presenta mayores dificultades a los alumnos.

Durante nuestros años como estudiantes, nos hemos enfrentado a diferentes maneras de aprendizaje de las matemáticas. Algunas de ellas eran demasiado teóricas, lo que dificultaba la comprensión y la posterior aplicación práctica, además de disminuir la motivación.

Bajo nuestro punto de vista, es importante que los alumnos sepan cuáles son los temas que se van a estudiar a lo largo del curso y para qué servirá cada uno de ellos, pero sobre todo, cuál será la utilidad en la vida diaria. Además el uso de ordenadores en las clases facilita el estudio de las matemáticas ya que se gana en motivación por parte de los alumnos y ayuda con la visualización de gráficos y el ahorro de tiempo en los cálculos.

Es por todo ello que el presente trabajo pretende diseñar y desarrollar una herramienta que permita la enseñanza de las matemáticas de una manera interactiva y entretenida donde se permite a los profesores registrados compartir material educativo con sus alumnos y ver una recreación de lo que cada uno de ellos ha hecho para así poder analizar cómo ha resuelto cada problema cada alumno, además de obtener diversos datos estadísticos sobre la evolución de los alumnos.

Además, a los alumnos se les plantearán los problemas que hayan definido los profesores y se les facilitarán herramientas de apoyo. La herramienta básica de la aplicación es una calculadora, donde dependiendo del problema planteado se inhabilitarán algunas teclas para forzar que el problema se resuelva mediante determinadas técnicas.

1.2 OBJETIVOS E HIPÓTESIS DE TRABAJO

Actualmente existen muchas páginas web que realizan algún tipo de apoyo a la enseñanza de las matemáticas. Sin embargo, si entramos en el campo de las aplicaciones con registro tanto para alumnos como para profesores se encuentran todavía en fase de desarrollo lo que hace que existan muy pocas herramientas para que puedan ser usadas por el profesional de la enseñanza.

La falta de herramientas de este tipo se debe al hecho de que no es una tarea fácil ya que existen muchos factores que dificultan el proceso, tales como: llevar un registro total de los profesores y alumnos que utilizan la aplicación, la recreación de todos los pasos dados por el alumno y sobre todo la incorporación de una singular calculadora.

Teniendo en cuenta estas hipótesis de partida, se plantean dos tipos de objetivos para el presente trabajo, los objetivos software que se pretendían alcanzar y los objetivos que se pretendían alcanzar a nivel personal.

1.2.1 *Objetivos software*

En líneas generales y como objetivo principal, se pretende desarrollar una aplicación web que cumpla los servicios de enseñanza para alumnos de primaria. Una vez fijados los puntos indispensables e imprescindibles de este proyecto, se extraen los siguientes objetivos software:

- **Gestión de usuarios:** Se llevará a cabo un registro de los diferentes clientes que accedan a la aplicación diferenciando tres tipos de usuarios: profesor, alumno y administrador, cada uno de ellos con sus permisos y privilegios.
- **Gestión de servicios:** Referentes a los diferentes servicios que la aplicación ofrece a los usuarios.
- **Gestión de historial de progresos:** Se tendrán que almacenar todos los pasos que un alumno de a la hora de resolver los problemas propuestos.
- **Gestión de calificaciones:** Para cada problema que resuelva un alumno, el profesor podrá darle una valoración numérica y escribir comentarios acerca de la resolución del mismo. Además, se podrá calcular la media aritmética con las calificaciones ya otorgadas.
- **Gestión de foro:** La aplicación dispondrá de un foro para que los usuarios registrados intercambien ideas y/o dudas sobre los ejercicios propuestos.
- **Gestión de búsquedas de ejercicios y de alumnos:** Se proporcionará a los profesores herramientas para la búsqueda de alumnos y problemas registrados en la aplicación.

A partir de estos objetivos principales y otros desglosados en el Capítulo 4, se trata de conseguir el objetivo final, una herramienta de apoyo a la enseñanza de matemáticas en educación primaria.

1.2.2 *Objetivos personales*

Con este Proyecto de Fin de Carrera, hemos intentado, y conseguido, alcanzar una serie de objetivos a nivel personal. El desarrollar este proyecto nos ha hecho adentrarnos en los conceptos del desarrollo web. En particular, los objetivos que hemos abordado y satisfecho son:

- Poder poner en práctica los conocimientos adquiridos durante la carrera de Ingeniería Informática.
- Aprender y profundizar en las técnicas de programación web.
- Perfeccionar el conocimiento de SQL y de los lenguajes de programación PHP, HTML y JavaScript.
- Ser capaces de enfrentarnos a un proyecto software desde el principio hasta el final abordando todos los campos de ingeniería del software previamente aprendidos durante la carrera.

1.3 **SOLUCIÓN PLANTEADA**

La solución planteada en el presente trabajo es una aplicación web que permite la enseñanza de matemáticas a alumnos de educación primaria de manera divertida, promoviendo un aprendizaje innovador.

Las técnicas utilizadas para ello incluyen herramientas de apoyo, que junto con varios tipos de materiales didácticos y un proceso final de calificación por parte de los profesores, hacen posible la motivación necesaria que ayude a los alumnos en sus estudios.

La potencia del método se ve incrementada por la ausencia de cualquier tipo de penalización en caso de no realizar correctamente los ejercicios, dando la posibilidad de repetirlos, y por premiar su correcta realización con juegos matemáticos que recompensen el esfuerzo realizado a la vez que se amplía el conocimiento.

Además al profesor se le permite recrear los pasos dados por el alumno en la resolución de los problemas, para ayudarle a conferir una calificación más exacta según el trabajo realizado y ofrecer ayuda más específica al alumno corrigiendo sus errores.

Aunque los resultados obtenidos pueden recuperarse en un documento de texto de uso general, se ofrece también un visor dentro de la aplicación que permite obtener tanto los resultados como los datos del alumno. Todo ello facilitará el trabajo del profesor, permitiéndole procesar las calificaciones, guardarlas y abrir los resultados obtenidos de sus alumnos desde su propio ordenador, sin necesidad de entrar en la aplicación.

1.4 ESTADO DEL ARTE

En este apartado se referencia el campo sobre el que vamos a aplicar el proyecto, las herramientas existentes en este ámbito así como también las herramientas que nos servirán de ayuda para el desarrollo del presente trabajo.

En un sentido amplio de la palabra se define una herramienta de apoyo como aquel elemento elaborado con el objetivo de hacer más sencilla una determinada actividad a través de una enseñanza alegre, aplicada e interactiva de diversas disciplinas y aportar la complementación y apoyo mediante esquemas no formales de aprendizaje. En nuestro caso, orientado hacia la enseñanza de matemáticas.

1.4.1 *Software educativos de referencia*

Consideramos el software educativo como un programa cuya función principal es el apoyo a la enseñanza y el aprendizaje, que junto con la introducción de la tecnología en la educación ha cambiado el modelo de enseñanza pasando de una enseñanza centrada en el profesor hacia un aprendizaje donde el protagonista es el alumno apoyado en una plataforma tecnológica.

En esta parte del trabajo hemos hecho unas indagaciones previas que sustentan el estudio, tratan sobre el mismo problema o se relacionan con otros. Nos servirán de guía y permitirán hacer comparaciones y tener ideas sobre cómo se trató el problema en casos anteriores. (Véase [7] y [8])

❖ **PDI, el libro digital**

Muestra los contenidos de forma paginada (PDF), ofrece algunas interacciones y enlaces y además aporta herramientas de zoom y paginación. Son proporcionados por la editorial. Facilita la rápida corrección de ejercicios.

También incluye herramientas de la pizarra como los transportadores, compás, reglas,... Que permiten realizar tareas como medir ángulos, medir segmentos, trazar líneas, dibujar arcos o circunferencias, etc. Como paso previo es necesario capturar la página del libro digital.

❖ **Control remoto con RealVNC**

Desde el ordenador del profesor se puede actuar sobre un equipo del alumno sin necesidad de tener que acudir físicamente al pupitre del mismo. Esto se consigue mediante la utilización de un software de control remoto de aula. En el mercado hay distintas aplicaciones de control remoto comerciales. Entre ellas destaca:

Este programa puede realizar asistencia remota en el ordenador del alumno y además puede mostrar el escritorio del alumno en el video-proyector a través del ordenador del profesor. Esta solución funciona perfectamente en la red local Wifi que forman los ordenadores del alumnado y profesorado.

❖ **Intercambio de archivos**

Consiste en la organización de archivos en el equipo del alumno. Después debe enviárselo al tutor por una de las siguientes maneras:

- Envío al tutor como adjunto por email. Ideal para archivos de reducido peso.
- El alumno guarda el archivo en su ordenador y luego en la memoria USB. Entrega al profesor el USB debidamente etiquetado con su nombre. Ideal para archivos de cierto peso, tales como, presentaciones, imágenes, vídeos, audios, etc.
- En un servidor de archivos: Mowes+exTplorer, que permite subir y descargar archivos desde el equipo del alumno a través del navegador web. Todo se guarda en el ordenador del profesor. Funciona en red local sin acceso a Internet.

❖ **Multimedia interactivos**

- **Paquetes JClic:** Permiten el repaso de contenidos de cálculo, geometría, etc. Se integran fácilmente en un curso de Moodle.
- **Cálculo con MATBIT:** Programa para practicar el cálculo en las cuatro operaciones básicas. Software de Vicens-Vives. Tiene autocorrección y distintos niveles de dificultad.

❖ **Aula virtual Moodle**

Diseño de agendas de trabajo por tema y área. El alumnado accede a recursos informativos e interactivos. Cuenta con la posibilidad de enviar archivos al tutor, además de la comunicación por foros y mensajería interna. Se puede hacer un seguimiento tutorial del alumnado.

1.4.2. Conclusiones

Nuestro proyecto, al igual que el proyecto Moodle, el más avanzado en esta materia, también ayuda a facilitar la enseñanza de asignaturas, en concreto las matemáticas. La principal diferencia está en que el proyecto Moodle se parece más a un repositorio de información con intercambio de archivos y comunicación entre alumno y profesor mientras que el nuestro está orientado hacia una enseñanza lúdica que motive al alumno en sus estudios.

1.5 ORGANIZACIÓN DEL DOCUMENTO

- **Capítulo 1, Introducción:** En este capítulo inicial se plantea el tema que concierne al proyecto que se está documentando. Se realiza una descripción concisa y clara de la principal finalidad del proyecto, y su utilidad en el entorno actual. Se establecen cuáles son los objetivos fundamentales que la aplicación plantea, haciendo referencia, a modo de introducción, a los requisitos de software necesarios para el sistema. Con esta información conseguiremos comprender mejor la funcionalidad que tendrá la aplicación, y qué es lo que se ha buscado con el desarrollo de dicho proyecto software. Además, vamos a introducir algunas nociones referentes al campo sobre el que vamos a aplicar el proyecto y las herramientas existentes en este ámbito. También se detalla la estructura de la memoria, especificando en cada apartado qué se describe.

- **Capítulo 2, Técnicas y herramientas utilizadas:** En este capítulo vamos a describir las metodologías y herramientas utilizadas en el desarrollo de este proyecto, exponiendo sus características principales y justificando los motivos de su elección.

- **Capítulo 3, Estudio de viabilidad de software:** En este capítulo se efectúa un repaso a los diferentes problemas que pueden surgir antes, durante y tras la realización del software. Estas dificultades pueden encontrarse en diferentes ámbitos de creación y explotación del programa; por ello, es necesaria la definición de un marco donde se explique la situación que ha dado origen al software y su posible expansión. Además contendrá la planificación que se ha seguido para el desarrollo software realizado, estructurando dicha planificación de manera temporal. Se expondrán de manera gráfica, mediante un diagrama de Gantt, las distintas fases que han tenido lugar.

- **Capítulo 4, Aspectos relevantes del desarrollo:** En este capítulo se recoge una exposición del ciclo de vida utilizado y los detalles de mayor relevancia de la fase de análisis. Además se incluyen los aspectos más interesantes de la implementación, documentación y pruebas.

- **Capítulo 5, Especificación de requisitos del software:** En este capítulo vamos a hacer un recorrido por el proceso de especificación de requisitos. En él se especificarán con cierto nivel de detalle cada uno de los objetivos que se alcanzarán en el sistema, así como los requisitos funcionales, no funcionales y de información que definen la creación del software. Todo esto conllevará a la utilización de métodos como casos de uso que completarán las especificaciones detalladas con la utilización de las REM. Esta parte se denominará como una definición del Dominio del Problema.

- **Capítulo 6, Especificación de diseño del software:** En este capítulo se pasará de la parte del análisis a la resolución de dicho análisis mediante la fase de diseño, donde se comenzará a concretar cada una de las soluciones a los problemas planteados durante la recopilación de requisitos software.

- **Capítulo 7, Técnica de Programación:** Este capítulo contenido en la memoria del proyecto, nos servirá de guía para mostrar la implementación de nuestro sistema. Se especificarán los métodos y funcionalidades de las que se encarga cada módulo de implementación. Es un refuerzo para comprender mejor el código fuente de la aplicación de cara a futuros desarrolladores que necesiten nuestro sistema.

- **Capítulo 8, Funcionalidad de la aplicación:** En este capítulo se recogen las funcionalidades principales de la aplicación dependiendo de qué tipo de usuario esté haciendo uso de la misma: profesor o alumno. Puede servir de guía para el usuario del sistema, ya que aquí se detallarán los conceptos necesarios para el uso de la aplicación con imágenes de ejemplo y especificaciones de cada funcionalidad.

- **Capítulo 9, Conclusiones y líneas de trabajo futuras:** En este capítulo se incluyen las conclusiones que se han derivado del desarrollo del proyecto y funcionalidades que aún pueden aplicársele al sistema que se ha elaborado.

Capítulo 2

Técnicas y herramientas utilizadas

En este capítulo vamos a describir las metodologías y herramientas utilizadas en el desarrollo de este proyecto, exponiendo sus características principales y justificando los motivos de su elección.

2.1 METODOLOGÍAS

2.1.1 Proceso Unificado

El Proceso Unificado es un marco de desarrollo software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, por ser iterativo e incremental y estar enfocado en los riesgos.



Ilustración 1: Proceso Unificado

El Proceso Unificado no es simplemente un proceso, es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. (Véase [9])

Se ha escogido el Proceso Unificado como marco de desarrollo software por la gran facilidad que aporta a la introducción de nuevos requisitos casi en cualquier momento del desarrollo por su naturaleza iterativa e incremental.

2.1.2 Programación Orientada a Objetos

La programación orientada a objetos o POO es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. En la

actualidad, existe gran variedad de lenguajes de programación que soportan la orientación a objetos. En este caso se ha utilizado PHP para implementar este tipo de metodología. (Véase [10])

Se ha escogido esta técnica de programación por su facilidad para la reutilización y extensión del código, porque agiliza el desarrollo de software y facilita el mantenimiento del mismo.

2.1.3 Metodología de Durán y Bernárdez

Para la elección de requisitos, se siguió la metodología de Durán y Bernárdez, desarrollada en el departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla [1]. Se ha escogido esta metodología porque se puede integrar perfectamente en el proceso unificado, siendo de fácil uso, al estar dirigida por plantillas.

2.2 PROGRAMACIÓN

En la realización de este Proyecto de Fin de Carrera se han empleado los siguientes lenguajes:

2.2.1 PHP

El lenguaje PHP se define como interpretado (no requiere compilación) y de alto nivel. El código de sus instrucciones va insertado en páginas HTML y es interpretado en el servidor. Se trata de un lenguaje de estilo clásico, del lado del servidor, cercano en su sintaxis a C++. (Véase [11])



Ilustración 2: PHP

Se ha escogido PHP frente a otros lenguajes por su sencillez, tanto a la hora de instalar, como de desarrollar. Además soporta la orientación a objetos, por lo que se ajustaba a la idea de realizar la aplicación con programación orientada a objetos.

2.2.2 JavaScript

JavaScript es un lenguaje de programación interpretado utilizado principalmente en páginas web, con una sintaxis semejante a la de Java. Es un lenguaje orientado a objetos, pero basado en prototipos, no en clases. (Véase [12])



Ilustración 3: JavaScript

Durante el desarrollo de la aplicación, se ha utilizado principalmente para comportamientos de acción.

2.2.3 CSS

Las hojas de estilo en cascada (CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.



Ilustración 4: CSS

Se ha escogido utilizar CSS por la facilidad y comodidad que aportan al diseño de páginas web. CSS se ha utilizado para definir los estilos tanto de todas las páginas de la aplicación como de las diferentes secciones de las mismas.

2.2.4 HTML

HTML es un formato de datos que permite el intercambio de información a través del World Wide Web. Está basado en el uso de hipertextos y que utiliza el protocolo http como medio para publicar información. El HTML, pues, no puede definirse como un lenguaje de programación, sino como un sistema de marcado. (Véase [13])



Ilustración 5: HTML

Se ha escogido esta herramienta por ser ya conocida, por su facilidad de implementación y porque es un lenguaje universal que es usado por casi todos los portales web en la implementación de sus sitios web.

2.2.5 AJAX

Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. (Véase [14])



Ilustración 6: AJAX

De esta forma es posible realizar cambios sobre la misma página sin necesidad de refrescarla, lo que significa aumentar la interactividad, velocidad y usabilidad en la misma.

2.2.6 jQuery

jQuery es una biblioteca o framework gratuito de JavaScript, que permite la realización de programas JavaScript de una forma simple y sencilla, creando páginas Web de las aplicaciones dinámicas complejas.



Ilustración 7: jQuery

jQuery tiene diversas prestaciones, entre las que destacan: el control de navegador de usuario, que permite despreocuparse de la compatibilidad de los scripts con los distintos navegadores existentes; mayor facilidad en la creación de aplicaciones del lado cliente, es decir, interfaces de usuario, efectos dinámicos o aplicaciones que hacen uso de AJAX. (Véase [15])

2.3 SISTEMAS DE ALMACENAMIENTO

2.3.1 MySQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones, lo cual demuestra su robustez y fiabilidad.



Ilustración 8: MySQL

El lenguaje de consulta estructurado (SQL) es un lenguaje declarativo de acceso a base de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y del cálculo relacional. (Véase [16])

Se optó por este sistema gestor de bases de datos por ser libre, rápido y eficaz cuando se trabaja con un número de tablas no demasiado grande, como era el caso.

2.3.2 Servidor Web: Apache

Podríamos definir un servidor Web como una aplicación que permite acceder a los recursos contenidos en algunos de los directorios del ordenador que alberga a usuarios remotos que realizan sus peticiones mediante protocolo HTTP. (Véase [17]).



Ilustración 9: Apache

El servidor escogido ha sido Apache. Bajo este nombre suele hacerse referencia a Apache Software Foundation, organización norteamericana que se autodefine con el objetivo

de facilitar ayuda organizativa, legal y financiera para los proyectos de desarrollo de software tipo Open Source (código abierto). (Véase [18])

Ha sido escogido por haber trabajado con él con anterioridad y por estar incluido en la herramienta XAMPP que integra MySQL y Apache.

2.4 HERRAMIENTAS DE DESARROLLO

A continuación se pasa a describir el conjunto de herramientas utilizadas para llevar a cabo las técnicas antes descritas.

2.4.1 REM

Para la fase de requisitos se utilizó la herramienta REM, que permite gestionar la información generada y presentarla adecuadamente con relativa sencillez.



Ilustración 10: REM

Se utilizó esta herramienta por las características ya mencionadas en la metodología de Durán y Bernárdez.

2.4.2 Visual Paradigm for UML

Visual Paradigm for UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. (Véase [19])



Ilustración 11: Visual Paradigm for UML

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor coste.

Se ha elegido esta herramienta porque permite realizar todos los tipos de diagramas: de clases, de casos de uso, de secuencia, diagramas de paquetes, etc...

2.4.3 WampServer

WampServer es un servidor, de software libre, independiente de plataforma que consiste en el gestor de base de datos MySQL, el servidor web Apache y los intérpretes para los lenguajes PHP y Perl. (Véase [20])



Ilustración 12: WampServer

Se ha escogido esta herramienta por integrar las principales herramientas y lenguajes que han sido utilizados para el desarrollo de la aplicación.

2.4.4 phpMyAdmin

phpMyAdmin es un conjunto de utilidades y scripts escritos en lenguaje PHP que permiten gestionar bases de datos MySQL a través de una página web. (Véase [21])



Ilustración 13: phpMyAdmin

Se escogió esta herramienta por la comodidad y facilidad que aporta para modificar, consultar, crear y borrar tanto bases de datos como tablas y registros contenidos en ellas. Además de porque permite la gestión de usuarios (nombres y contraseñas) y establecer sus privilegios sobre la base de datos.

2.4.5 Dreamweaver

Es una aplicación en forma de estudio pero con más parecido a un taller destinado para la construcción y edición de sitios y aplicaciones Web basados en estándares. (Véase [22])



Ilustración 14: Dreamweaver

Se ha elegido este programa por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

2.4.6 MySQL Workbench

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. (Véase [23])



Ilustración 15: MySQL Workbench

2.4.7 Gantt Project

Gantt Project es una iniciativa de código abierto cuyo propósito es realizar representaciones esquemáticas de la distribución de las tareas de un proyecto a corto, medio o largo plazo. (Véase [24])



Ilustración 16: Gantt Project

Se ha usado para crear el diagrama de Gantt de la planificación temporal del proyecto. Este diagrama se encuentra en el Capítulo 3.

Se ha escogido este recurso frente a otros por ser libre y por tener conocimiento previo del manejo del mismo.

2.5 HERRAMIENTAS DE DOCUMENTACIÓN

2.5.1 Microsoft Word

Microsoft Word es un software destinado al procesamiento de textos. Con esta herramienta se han elaborado los documentos que forman la memoria. (Véase [25])



Ilustración 17: Microsoft Word

Las razones para escoger esta herramienta es la experiencia previa en el manejo de ésta.

2.6 HERRAMIENTAS DE DISEÑO GRÁFICO

2.6.1 Adobe Photoshop

Se trata de una aplicación informática en forma de taller de pintura y fotografía que trabaja sobre un “lienzo” y que está destinado para la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits. (Véase [26])



Ilustración 18: Adobe Photoshop

Se ha utilizado para editar las imágenes de la aplicación.

Capítulo 3

Estudio de viabilidad de software

En este capítulo se efectúa un repaso a los diferentes problemas que pueden surgir antes, durante y tras la realización del software. Estas dificultades pueden encontrarse en diferentes ámbitos de creación y explotación del programa; por ello, es necesaria la definición de un marco donde se explique la situación que ha dado origen al software y su posible expansión.

3.1 INTRODUCCIÓN

Existen multitud de guías en la que se explican los diferentes riesgos que pueden darse en la creación de software, por ello se toman distintos subconjuntos de riesgos conocidos y se realiza una evaluación de estos, enfocados en el desarrollo del proyecto de fin de carrera.

Se debe realizar una evaluación de los conocimientos técnicos necesarios para la realización de la aplicación. Una parte importante en la realización del proyecto será la adquisición de los conocimientos necesarios para su posterior codificación.

Además, desde un punto de vista objetivo, se debe realizar un repaso de algunas de las distintas alternativas que se pueden encontrar para realizar un servicio similar al que proporciona el software a desarrollar.

3.2 PLANIFICACIÓN TEMPORAL DEL PROYECTO

En este apartado se muestra una planificación del tiempo dedicado a cada una de las tareas dentro de la realización del Proyecto de Fin de Carrera. Para facilitar la comprensión de esta organización se utiliza un gráfico de Gantt.

En el eje horizontal se encuentra la escala temporal expresada en meses desde que se comienza el proyecto a la fecha de finalización. En el eje vertical se muestran las diferentes

tareas realizadas. Las barras relacionan las tareas realizadas en la relación al tiempo planificado.

		
Nombre	Fecha de inicio	Fecha de fin
Documentación PHP	17/09/12	30/11/12
Documentación SQL	8/10/12	30/11/12
Total documentación	17/09/12	30/11/12
Elicitación de requisitos	3/12/12	18/01/13
Análisis de software	20/12/12	30/01/13
Diseño del software	21/01/13	28/02/13
Desarrollo de la memoria	4/02/13	23/08/13
Codificación	28/02/13	30/08/13
Correcciones de la memoria	26/08/13	7/09/13

Ilustración 19: Fechas previstas para el proyecto

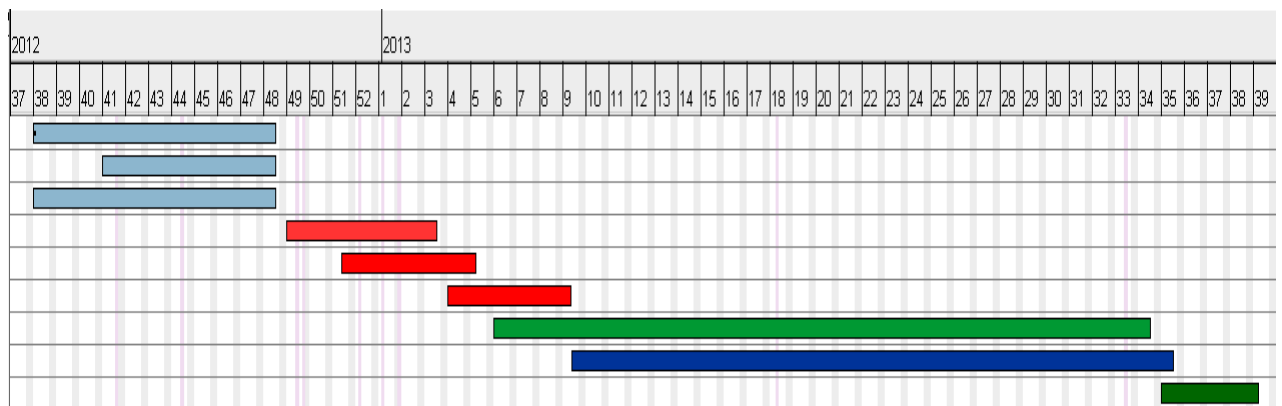


Ilustración 20: Diagrama de Gantt

3.3 EVALUACIÓN DEL RIESGO TÉCNICO

Para realizar una evaluación de riesgos que pueden producirse en la creación de la aplicación en este apartado se va a crear una lista de comprobación de elementos de riesgo. Esta lista se centra en subconjuntos de riesgos conocidos y predecibles dentro del desarrollo software.

- **Impacto en el negocio:** Riesgos asociados con las limitaciones impuestas por la gestión o por el mercado.

Como el desarrollo software de nuestro proyecto no está limitado a ningún impuesto de gestión o mercantil, el impacto en el negocio no debería ser un riesgo considerable. El hecho de que no se pretende la comercialización de la aplicación, evita esta clase de riesgos.

- **Características del cliente:** Riesgos asociados con la sofisticación del cliente y la habilidad del desarrollador para comunicarse con el cliente en los momentos oportunos.

El cliente, en este caso, son los profesores de la Facultad de Educación que solicitaron la realización de la aplicación. Esta posibilidad permitió fijar los objetivos con perspectiva. En caso de un cambio en los requisitos, habrá una reunión con el director del proyecto, al igual que de producirse un cambio en la manera de pensar del cliente se deberá notificar a los desarrolladores a través de él.

- **Entorno de desarrollo:** Riesgos asociados con la disponibilidad y calidad de las herramientas que se van a emplear en la construcción del producto. El funcionamiento de software está ligado a la calidad de las herramientas utilizadas en su desarrollo.

Para minimizar este riesgo, se utilizará, en la medida de lo posible, recursos que sean de código abierto o libres; así siempre estarán disponibles y, en la mayoría de los casos, serán de buena calidad.

- **Tamaño y experiencia de la plantilla:** Riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo.

Este riesgo debería quedar minimizado por el importante esfuerzo de documentación realizado antes del desarrollo. En caso de falta de información y gracias a que se trata de herramientas libres, se utilizará Internet como utilidad para encontrar blogs y foros especializados en el tema.

Capítulo 4

Aspectos relevantes del desarrollo

En este capítulo se recogen una exposición del ciclo de vida utilizado y los detalles de mayor relevancia de la fase de análisis. Además se incluyen los aspectos más interesantes de la implementación, documentación y pruebas.

4.1 CICLO DE VIDA

Al planificar un proyecto, uno de los puntos más importantes y relevantes es decidir qué pautas seguirá el trabajo a realizar e intentar acertar con la elección. En este caso, como marco de trabajo de desarrollo del proyecto se ha escogido el Proceso Unificado. (Véase [31])

Es un marco de desarrollo iterativo e incremental que presenta las siguientes características:

- **Basado en componentes:** El sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.
- **Conducido por casos de uso:** Aseguran que toda la funcionalidad se realiza en el sistema y verifican y prueban el mismo.
- **Centrado en la arquitectura:** Permitirá describir mediante vistas el sistema y establecer una arquitectura básica pronto (en este caso esto es algo sumamente importante debido a las limitaciones temporales del Proyecto de Fin de Carrera).
- **Iterativo e incremental:** Se basa en la ampliación y el refinamiento del sistema mediante una serie de desarrollos cortos que no sólo mejoran el sistema sino que también lo hacen crecer.

4.2 ANÁLISIS

El análisis es la parte del proceso que tiene como principal propósito realizar un modelo de dominio del problema para ofrecer una especificación más precisa de los requisitos.

Para la especificación de requisitos del software (ERS) se ha tomado como herramienta la metodología de Durán y Bernárdez. Con esta metodología pretendemos obtener y documentar los objetivos a partir de las plantillas que este método propone para obtener como resultado final el modelo de casos de uso.

Las principales tareas dentro del proceso de análisis se muestran a continuación, incluyendo el contexto del propio proyecto:

- **Obtener información sobre el dominio del problema y del sistema actual:** Se basa principalmente en conocer el dominio del problema y la situación actual del mismo.

En este caso, Calcul@ es una herramienta de apoyo a la enseñanza de matemáticas en educación primaria dirigida a niños entre 6 y 12 años. Para comenzar su desarrollo fue necesario establecer una entrevista previa con el cliente, para este proyecto en particular un profesor de Didáctica de las Matemáticas de la Universidad Complutense de Madrid, en ella establecimos las condiciones iniciales y los requisitos básicos de la futura aplicación.

Este es un punto muy importante en el proceso ya que no se puede olvidar que, al fin y al cabo, el producto está dirigido a un cliente y es el cliente quien tiene que quedar satisfecho con el resultado final.

- **Identificar/revisar los objetivos del sistema:** Este es un punto fundamental dentro del proceso de análisis.

En nuestro caso, los objetivos principales son la gestión de usuarios, de ejercicios, llevar las calificaciones de cada usuario y un historial de los pasos realizados al solucionar los ejercicios. Luego se fueron añadiendo más objetivos como el foro.

- **Identificar/revisar los requisitos de información:** El objetivo de esta tarea es identificar los requisitos de almacenamiento de la información que ha de cumplir el sistema software a desarrollar.

De manera rápida, se puede decir que la información que ha de almacenar el sistema es: todos los datos personales de los usuarios, problemas, calificaciones, cursos, pasos realizados por los alumnos al resolver los ejercicios y posts del foro.

- **Identificar/revisar los requisitos funcionales:** El objetivo de esta tarea es identificar los actores del sistema, identificar los requisitos funcionales, expresados de forma tradicional o como casos de uso, que deberá cumplir el sistema software que se está desarrollando.
- **Identificar/revisar los requisitos no funcionales:** El objetivo de este trabajo es identificar los requisitos no funcionales del sistema. Requisitos indispensables de este tipo son eficiencia, accesibilidad, usabilidad y portabilidad.

Gracias a las plantillas, se han definido una serie de requisitos (tanto funcionales como no funcionales), objetivos, actores y las principales funcionalidades del sistema como recogeremos en el Capítulo 5.

4.3 ENTORNO TECNOLÓGICO

En este apartado se describe el entorno en el que se ha construido este sistema. Está dividido en dos partes: entorno software y entorno hardware.

4.3.1 Entorno software

El desarrollo se ha llevado a cabo en Windows 7 Professional con sistema operativo de 32 bits. Se eligió este sistema por su robustez, su compatibilidad con los servidores y por la multitud de software para el desarrollo web disponible para esta plataforma.

Como lenguaje de programación se ha elegido PHP (Versión 5.3.5) por ser un lenguaje interpretado, creado específicamente para la creación de páginas web dinámicas, por ser libre. Esto quiere decir que en su desarrollo hay un proceso de colaboración que hace que se tenga inmediatamente disponible, y de forma gratuita, una enorme cantidad de recursos: el lenguaje en sí, el servidor para ejecutarlo, manuales y tutoriales sin fin, y sobre todo scripts que puedes descargar y usar. Esta abundancia de código libremente disponible ayuda aún más en el proceso de aprendizaje. Además se ha completado con el uso de JavaScript, hojas de estilo CSS y como lenguaje de marcado HTML.

Para trabajar de forma local, se ha escogido WAMPSEVER (Versión 2.1) que es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL (Versión 5.5.8), el servidor Web Apache (Versión 2.2.17) y los intérpretes para lenguajes de script: PHP y Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Se ha escogido esta herramienta porque ya la conocíamos previamente y se ajustaba perfectamente a las necesidades del proyecto.

Como sistema gestor de base de datos, se ha utilizado MySQL (Versión 5.5.8), porque es libre y relacional aunque también por comodidad porque ya viene integrado en

WAMPSEVER. También se ha utilizado phpMyAdmin (Versión 3.3.9) que también viene integrado en dicha herramienta.

Macromedia Dreamweaver CS6 fue el entorno elegido para el desarrollo de las actividades de implementación. Es la herramienta de desarrollo Web líder del mercado y permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basadas en normas. Es un editor de HTML visual, diseñado para desarrolladores profesionales.

Como parte de la accesibilidad, se usaron los navegadores web más conocidos para comprobar que el sistema se ejecutaba correctamente en todos ellos. Concretamente Internet Explorer 9, Firefox 12.0, y Google Chrome 25.0 en Windows.

4.3.2 Entorno hardware

Para la implementación del sistema se han usado varios portátiles ASUS con procesador Core 3 Duo de Intel a 2.27 GHz y 4GB de memoria RAM.

En estos ordenadores se han llevado a cabo la programación y la confección de la memoria de este proyecto.

4.4 IMPLEMENTACIÓN

A la hora de la implementación, se han intentado llevar a cabo aspectos como la consistencia o la extensibilidad, pero sin dejar de pensar en la usabilidad o accesibilidad, objetivos primordiales desde la especificación.

En nuestro caso, teniendo en cuenta que está orientado a niños, la selección debe ser lo más dinámica posible, ya que una lista larga de posibilidades puede aburrir al niño, produciendo un efecto contrario al que se desea conseguir.

Como lenguaje de programación se eligió PHP ya que es un lenguaje muy ágil y robusto que permite crear aplicaciones web rápidamente ya que cuenta con múltiples componentes. Además se integra muy bien con Apache y MySQL, y cuenta con un API para trabajar con MySQL muy estable.

Los archivos principales y el sistema de clases utilizado, en el que se basa toda la aplicación y que dan sentido y soporte a la misma se pueden resumir en:

- **Archivo de configuración:** Donde está la configuración principal de la web y de su interfaz.
- **Archivo plantilla:** En los que se encuentran los principales archivos de inclusión y funciones importantes del desarrollo como las cabeceras o la validación de formatos.
- **Archivo de funciones:** Principales funciones de la base de datos de la aplicación.

4.5 **DOCUMENTACIÓN**

La documentación en un proyecto de software es importante porque permite conservar la historia, facilita la utilización por parte del usuario, garantiza la permanencia y disminuye los costos de operación y de ejecución del proyecto como tal.

La documentación de los programas es un aspecto sumamente importante, tanto en el desarrollo de la aplicación como en el mantenimiento de la misma. Al no hacer esta parte del desarrollo, se pierde la posibilidad de la reutilización de parte del software en otras aplicaciones.

La documentación de un programa empieza a la vez que la construcción del mismo y finaliza justo antes de la entrega del programa o aplicación al cliente, de hecho, en la realización de este proyecto, la documentación se llevó a cabo desde el principio de la elección de requisitos. (Véase [32])

La estructura de la documentación del Proyecto de Fin de Carrera se muestra en la introducción de este documento.

4.6 **PRUEBAS**

Este proyecto se ha llevado a cabo siguiendo un ciclo de vida iterativo-incremental, y como tal, las pruebas del sistema se han realizado al final de cada iteración.

Al finalizar cada pase por la implementación, se realizaron pruebas para comprobar si la nueva parte que se había añadido funcionaba sin producir errores y cumplía con los objetivos marcados en el diseño. Cada vez que se terminaba un incremento, además de comprobarse que funcionaba correctamente y sin errores, se realizaron pruebas de integración para verificar que no había ningún conflicto con el resto de partes de la aplicación y en conjunto, el sistema funcionaba como se esperaba. Finalmente, cuando se terminó la aplicación, se realizaron pruebas del sistema completo para comprobar que cumplía con todos los requisitos deseados, funcionaba correctamente y detectar algún posible error que pudiera haber. Se han realizado pruebas en distintos navegadores y en distintos sistemas operativos.

Estas pruebas suponían puntos de control y verificación para el proyecto, en especial, para saber si se cumplían o no los objetivos propuestos en la fase de análisis.

Capítulo 5

Especificación de requisitos del software

En este capítulo vamos a hacer un recorrido por el proceso de especificación de requisitos.

5.1 OBJETIVOS DEL PROYECTO

Para la definición de los objetivos del proyecto se utilizarán las plantillas definidas por el método de Durán y Bernárdez.

OBJ-0001	Sistema de registro y autenticación de profesores
Versión	1.0
Descripción	El sistema deberá mantener un sistema a través del cual los profesores puedan registrarse en la aplicación y autenticarse para entrar en ella y acceder a sus servicios.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0002	Gestión de alumnos
Versión	1.0
Descripción	El sistema deberá gestionar los datos de los alumnos registrados en la aplicación, gestionando cualquier modificación que se realice sobre ellos.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0003	Gestión de ejercicios
Versión	1.0
Descripción	El sistema deberá gestionar la información relativa a los ejercicios dados de alta por los profesores y realizados por los alumnos.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0004	Gestión de las calificaciones
Versión	1.0
Descripción	El sistema deberá gestionar la contabilidad de las calificaciones que los alumnos reciben por la realización de los objetivos.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0005	Herramienta de búsqueda de alumnos
Versión	1.0
Descripción	El sistema deberá proporcionar a los profesores una herramienta de búsqueda de alumnos para que sea más sencillo localizar el alumno del que deseamos saber los datos.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0006	Historial de pasos realizados por el alumno en la realización de ejercicios
Versión	1.0
Descripción	El sistema deberá mantener un historial de cada alumno donde se almacene la información de qué pasos ha dado el alumno al realizar los ejercicios propuestos por el profesor.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0007	Foro
Versión	1.0
Descripción	El sistema deberá contar con un foro donde alumnos y profesores puedan escribir e intercambiar sus dudas sobre los ejercicios.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Hay presión
Estado	Validado
Estabilidad	Media
Comentarios	Ninguno

OBJ-0008	Matricular cursos
Versión	1.0
Descripción	El sistema deberá contar con un sistema en el que los profesores puedan matricular cursos en los que tengan docencia.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0009	Matricular alumnos
Versión	1.0
Descripción	El sistema deberá contar con un sistema de registro en el que los profesores puedan matricular a los alumnos correspondientes a cada curso.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0010	Introducir problemas
Versión	1.0
Descripción	El sistema deberá contar con un sistema en el que los profesores puedan introducir problemas para que los alumnos los resuelvan.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0011	Autenticación de alumnos
Versión	1.0
Descripción	El sistema deberá mantener un sistema a través del cual los alumnos puedan autenticarse para entrar en la aplicación y acceder a sus servicios.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0012	Matricular profesores
Versión	1.0
Descripción	El sistema deberá contar con un sistema de registro en el que el administrador pueda matricular a los profesores que deseen utilizar la aplicación.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0013	Borrar profesores
Versión	1.0
Descripción	El sistema deberá contar con un sistema de borrado en el que el administrador pueda eliminar a los profesores que ya no deseen utilizar la aplicación.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

OBJ-0014	Gestión de profesores
Versión	1.0
Descripción	El sistema deberá contar con un sistema de gestión de profesores en el que el administrador pueda actualizar a los profesores dados de alta anteriormente en la aplicación.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

5.2 CATÁLOGO DE REQUISITOS**5.2.1 Requisitos de información**

IRQ-0001	Alumnos
Versión	1.0
Dependencias	[OBJ-0002] Gestión de alumnos [OBJ-0004] Gestión de calificaciones [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [OBJ-0007] Foro [OBJ-0011] Autenticación de alumnos [OBJ-0012] Agenda – calendario [UC-0006] Consultar datos alumno [UC-0010] Consultar problema [UC-0013] Resolver problema [UC-0014] Comentar foro [UC-0015] Leer foro
Descripción	El sistema deberá almacenar la información correspondiente a los datos personales de los alumnos registrados de la aplicación así como la puntuación que obtienen y los pasos dados en la resolución de los ejercicios realizados.
Datos específicos	Identificador, nombre, apellidos, teléfono, fecha de nacimiento, el nombre de usuario y el password que usan para acceder al sistema, sexo, el curso y la letra del mismo en el que está matriculado, las calificaciones, los pasos dados y la fecha de realización para cada ejercicio.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

IRQ-0002	Profesores
Versión	1.0
Dependencias	[OBJ-0001] Sistema de registro y autenticación de profesores [OBJ-0003] Gestión de ejercicios [OBJ-0004] Gestión de calificaciones [OBJ-0005] Herramienta de búsqueda de alumnos [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [OBJ-0007] Foro [OBJ-0008] Matricular cursos [OBJ-0009] Matricular alumnos [OBJ-0010] Introducir problemas [OBJ-0012] Agenda - calendario [UC-0001] Buscar alumno [UC-0002] Alta alumno [UC-0003] Baja alumno [UC-0004] Borrar datos del alumno [UC-0005] Modificación datos del alumno [UC-0006] Consultar datos alumno [UC-0007] Alta profesor [UC-0008] Alta problema [UC-0009] Eliminar problema [UC-0010] Consultar problema [UC-0011] Buscar problema [UC-0014] Comentar foro [UC-0015] Leer foro [UC-0016] Borrar comentarios foro
Descripción	El sistema deberá almacenar la información correspondiente a los datos personales de los profesores registrados en la aplicación.
Datos específicos	DNI, nombre, apellidos y el password que usan para acceder al sistema.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

IRQ-0003	Administrador
Versión	1.0
Dependencias	[OBJ-0012] Matricular profesores [OBJ-0013] Borrar profesores [OBJ-0014] Gestión de profesores
Descripción	El sistema no almacenará ningún tipo de información del administrador del sistema.
Datos específicos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

IRQ-0003	Problemas
Versión	1.0
Dependencias	[OBJ-0003] Gestión de ejercicios [OBJ-0004] Gestión de calificaciones [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [OBJ-0010] Introducir problemas [UC-0008] Alta problema [UC-0009] Eliminar problema [UC-0010] Consultar problema [UC-0011] Buscar problema [UC-0013] Resolver problema
Descripción	El sistema deberá almacenar la información correspondiente a los problemas ofrecidos por los profesores a sus alumnos.
Datos específicos	Identificador, nombre, descripción, ayuda teórica y tema al que pertenece.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

IRQ-0004	Historial
Versión	1.0
Dependencias	[OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [UC-0012] Almacenar datos en historial
Descripción	El sistema deberá almacenar la información correspondiente a los pasos realizados por cada alumno en la resolución de los ejercicios.
Datos específicos	Usuario que realiza el ejercicio, fecha en la que se realiza y los pasos dados en su resolución.
Importancia	Importante
Urgencia	Inmediatamente

Estado	Validado
Estabilidad	Alta

IRQ-0005	Calificaciones
Versión	1.0
Dependencias	[IRQ-0001] Alumnos [IRQ-0002] Profesores [OBJ-0004] Gestión de calificaciones
Descripción	El sistema deberá almacenar la información correspondiente a las calificaciones que cada profesor le da a cada uno de sus alumnos.
Datos específicos	Ninguno
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

IRQ-0006	Foro
Versión	1.0
Dependencias	[OBJ-0007] Foro [UC-0014] Comentar foro [UC-0015] Leer foro [UC-0016] Borrar comentarios foro
Descripción	El sistema deberá almacenar toda la información correspondiente al foro.
Datos específicos	Ninguno
Importancia	Importante
Urgencia	Hay presión
Estado	Validado
Estabilidad	Alta

5.3.2 Requisitos funcionales

La especificación de este tipo de requisitos se realizará mediante los diagramas de casos de uso.

5.3.2.1 Definición de actores

ACT-0001	Profesor
Versión	1.0
Descripción	Este actor representa a los profesores que usan la aplicación.
Comentarios	Ninguno

ACT-0002	Alumno
Versión	1.0 (11/02/2013)
Descripción	Este actor representa a los alumnos que han sido dados de alta por un profesor y pueden usar la aplicación.
Comentarios	Ninguno

ACT-0003	Administrador
Versión	1.0 (11/02/2013)
Descripción	Este actor representa a los usuarios de la aplicación que son administradores. Es una especificación de Profesor.
Comentarios	Ninguno

5.3.2.2 Diagramas de caso de uso

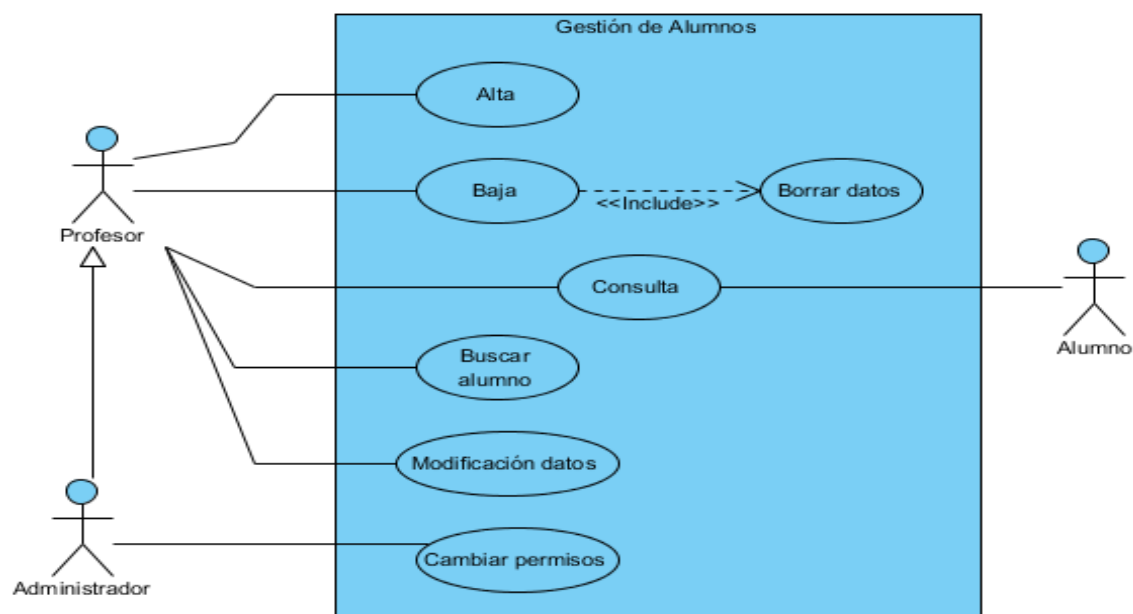


Ilustración 21: Diagrama de casos de uso. Gestión de alumnos

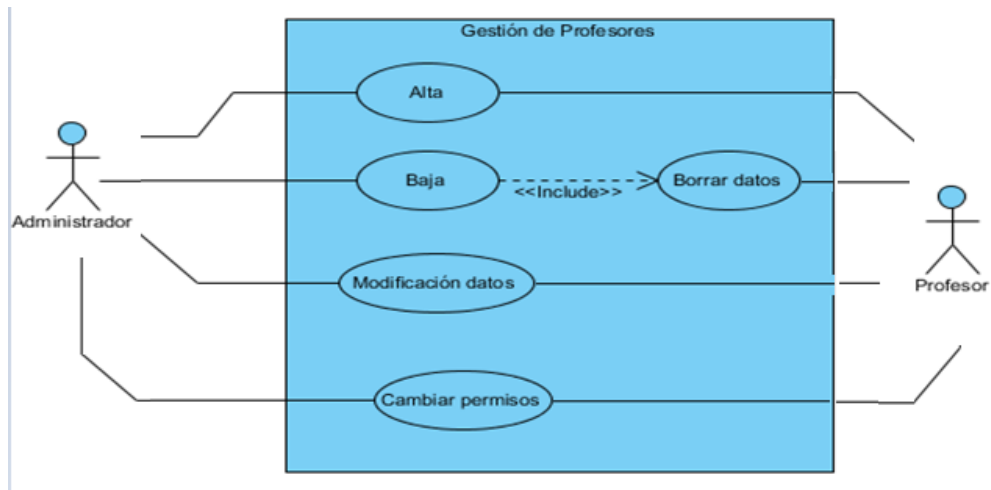


Ilustración 22: Diagrama de casos de uso. Gestión de profesores

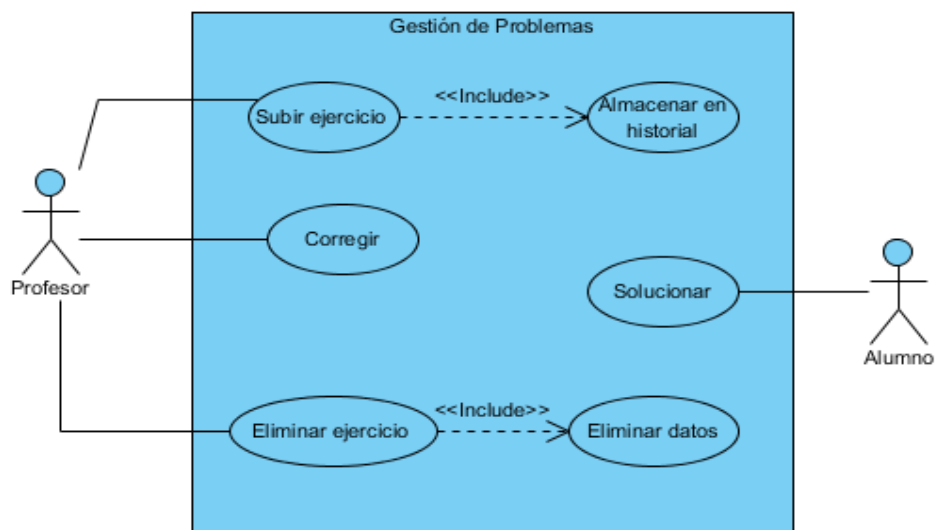


Ilustración 23: Diagrama de casos de uso. Gestión de problemas



Ilustración 24: Diagrama de casos de uso. Gestión de calificaciones

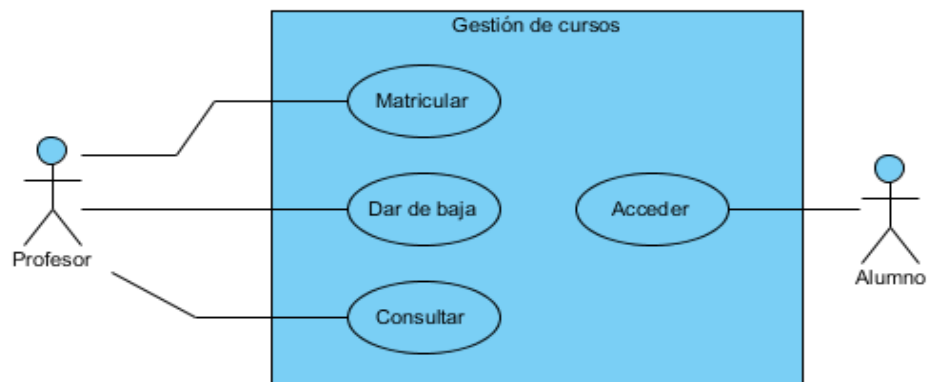


Ilustración 25: Diagrama de casos de uso. Gestión de cursos

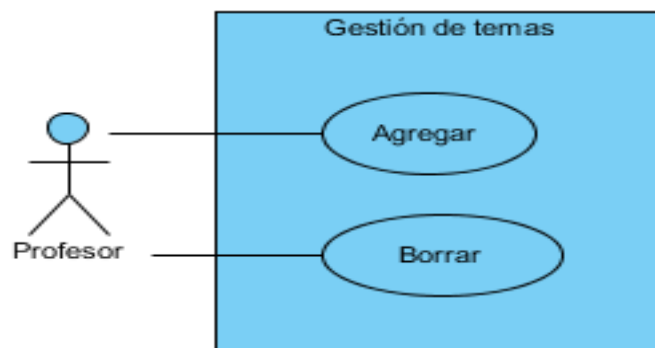


Ilustración 26: Diagrama de casos de uso. Gestión de temas

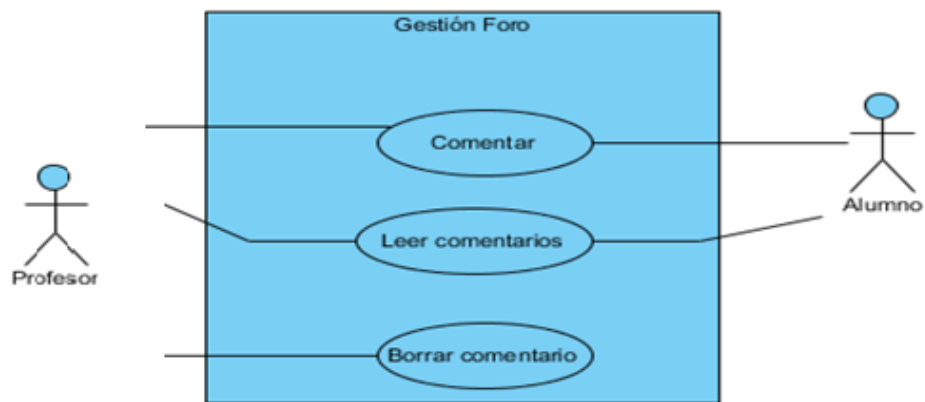


Ilustración 27: Diagrama de casos de uso. Gestión de foro

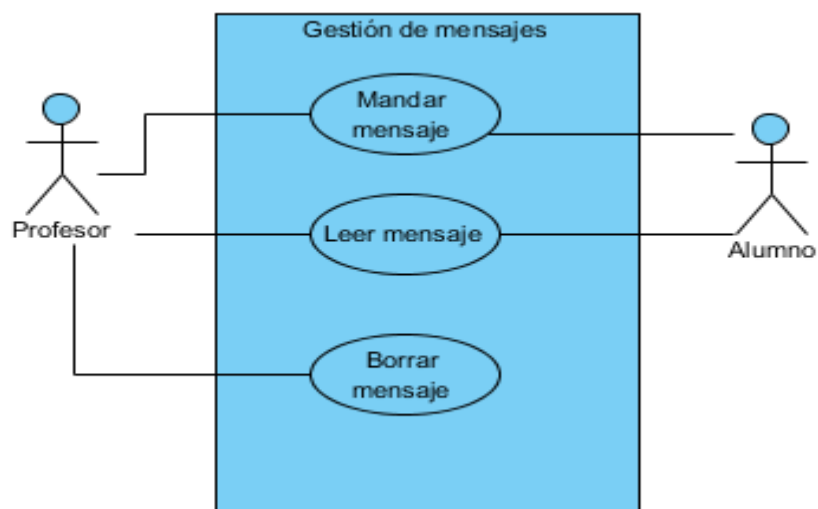


Ilustración 28: Diagrama de casos de uso. Gestión de mensajes

5.3.2.3 Casos de uso

UC-0001	Buscar alumno	
Versión	1.0	
Dependencias	[OBJ-0002] Gestión de alumnos [UC-0002] Alta alumno [UC-0006] Consulta datos alumno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee buscar un alumno en la base de datos.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) busca el alumno que desea buscar en la tabla de alumnos de cada curso.
Postcondición	El usuario que está realizando la búsqueda ha obtenido datos del usuario buscado; si este existe.	
Frecuencia esperada	2 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0002	Alta alumno	
Versión	1.0	
Dependencias	[OBJ-0002] Gestión de alumnos [OBJ-0011] Registro de alumnos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee que un alumno empiece a usar la aplicación.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) introduce los datos necesarios para dar de alta al actor Alumno (ACT-0002) en la aplicación.
Postcondición	Esa persona ha conseguido acceso a la aplicación como alumno.	
Frecuencia esperada	3 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0003	Baja alumno	
Versión	1.0	
Dependencias	[UC-0004] Borrar datos usuario [OBJ-0002] Gestión de alumnos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee dar de baja a un alumno.	
Precondición	El usuario debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) realiza los pasos necesarios para dar de baja al actor Alumno (ACT-0002) en la aplicación.
	2	Se realiza el caso de uso Borrar datos del usuario (UC-0004).
Postcondición	Todos los datos del usuario borrado han sido eliminados del sistema.	
Frecuencia esperada	1 vez por curso.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0004	Borrar datos del alumno	
Versión	1.0	
Dependencias	[OBJ-0002] Gestión de alumnos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso se dé de baja a un alumno o durante la realización de los siguientes casos de uso: [UC-0003] Baja usuario.	
Precondición	Se ha realizado el caso de uso [UC-0003] Baja usuario.	
Secuencia normal	Paso	Acción
	1	El sistema elimina cualquier dato del usuario que es dado de baja.
Postcondición	Todos los datos del usuario han sido eliminados del sistema.	
Frecuencia esperada	1 vez por curso.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0005	Modificación datos del alumno	
Versión	1.0	
Dependencias	[IRQ-0001] Alumnos [OBJ-0002] Gestión de alumnos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee modificar alguno de los datos de sus alumnos.	
Precondición	El usuario debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) modifica os datos guardados del actor Alumno (ACT-0002) en la aplicación.
Postcondición	Los datos del usuario han quedado correctamente modificados.	
Frecuencia esperada	1 vez cada 3 meses.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0006	Consulta datos alumnos	
Versión	1.0 (01/03/2013)	
Dependencias	[IRQ-0001] Alumnos [OBJ-0002] Gestión de alumnos [IRQ-0003] Historial [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [UC-0016] Almacenar datos en historial	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor consulte los datos guardados en la aplicación de uno de los alumnos.	
Precondición	El usuario debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) consulta los datos guardados del actor Alumno (ACT-0002) en la aplicación.
Postcondición	El profesor ha obtenido los datos buscados.	
Frecuencia esperada	1 vez cada 2 días.	
Importancia	Importante	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	

UC-0007	Alta profesor	
Versión	1.0	
Dependencias	[OBJ-0001] Sistema de registro y autenticación de profesores	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee empezar a usar la aplicación.	
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0003) introduce los datos necesarios para darse de alta en la aplicación.
Postcondición	Esa persona ha conseguido el acceso a la aplicación como profesor.	
Frecuencia esperada	1 vez por mes(es)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0008	Alta problema	
Versión	1.0	
Dependencias	[IRQ-0002] Profesores [OBJ-0003] Gestión de ejercicios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee dar de alta un ejercicio en la aplicación.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) introduce la información necesaria sobre un problema que desea empezar a ofrecer.
Postcondición	El ejercicio ha sido introducido en el sistema.	
Frecuencia esperada	4 veces por semana(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0009	Eliminar problema	
Versión	1.0	
Dependencias	[IRQ-0002] Profesores [OBJ-0003] Gestión de ejercicios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee dar eliminar un ejercicio en la aplicación.	
Precondición	El ejercicio debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) elimina el problema .
Postcondición	El ejercicio ha sido elimina en el sistema.	
Frecuencia esperada	1 vez por mes(es)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0010	Consultar problema	
Versión	1.0 (01/03/2013)	
Autores	Francisco Javier Heras Martín Silvia Panadero Hernández Estefanía Sánchez García-Velasco	
Dependencias	[IRQ-0001] Alumnos [IRQ-0002] Profesores [OBJ-0003] Gestión de ejercicios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor o alumno desee consultar un ejercicio en la aplicación.	
Precondición	El ejercicio debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) o el actor Alumno(ACT-0002) consulta el problema
Postcondición	El usuario ha obtenido la información buscada.	
Frecuencia esperada	5 veces por día(s)	
Importancia	Importante	
Urgencia	Hay presión	
Estado	Validado	
Estabilidad	Alta	

UC-0011	Buscar problema	
Versión	1.0	
Dependencias	[IRQ-0002] Profesores [OBJ-0003] Gestión de ejercicios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee buscar uno de los ejercicios dados de alta anteriormente en la aplicación.	
Precondición	El ejercicio debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) busca un problema
Postcondición	El usuario ha obtenido los datos del problema buscado.	
Frecuencia esperada	3 veces por día(s)	
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	

UC-0012	Almacenar datos en historial	
Versión	1.0	
Dependencias	[IRQ-0004] Historial [IRQ-0005] Calificaciones [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno resuelva los ejercicios y un profesor desee consultar el progreso de dicho alumno.	
Precondición	El ejercicio se ha resuelto.	
Secuencia normal	Paso	Acción
	1	El sistema almacena en el historial los pasos realizados por los alumnos que resuelven el ejercicio.
Postcondición	Los datos de la solución del ejercicio se han almacenado en el historial.	
Frecuencia esperada	10 veces por día(s)	
Importancia	Importante	
Urgencia	Hay presión	
Estado	Validado	
Estabilidad	Alta	

UC-0013	Resolver problema	
Versión	1.0	
Dependencias	[IRQ-0001] Alumnos [IRQ-0004] Historial [OBJ-0003] Gestión de ejercicios [OBJ-0006] Historial de pasos realizados por el alumno en la realización de ejercicios [UC-0012] Almacenar datos en historial	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno desee resolver uno de los problemas.	
Precondición	El ejercicio debe existir y debe estar disponible para ese alumno.	
Secuencia normal	Paso	Acción
	1	El actor Alumno (ACT-0002) resuelve un problema.
	2	Se realiza el caso de uso Almacenar datos en historial (UC-0016).
Postcondición	El alumno ha resuelto el ejercicio propuesto.	
Frecuencia esperada	10 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0014	Comentar foro	
Versión	1.0	
Dependencias	[IRQ-0005] Foro [OBJ-0007] Foro	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno o profesor desee comentar en el foro de la aplicación.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) o el actor Alumno (ACT-0002) escribe su comentario.
Postcondición	El comentario ha quedado registrado.	
Frecuencia esperada	2 veces por día(s)	
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	

UC-0015	Leer foro	
Versión	1.0 (01/03/2013)	
Dependencias	[IRQ-0005] Foro [OBJ-0007] Foro	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando tanto alumnos como profesores deseen leer comentarios del foro.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) o el actor Alumno (ACT-0002) entran y leen comentarios del foro.
Postcondición	El comentario ha quedado registrado.	
Frecuencia esperada	2 veces por día(s)	
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	

UC-0016	Borrar comentarios foro	
Versión	1.0 (01/03/2013)	
Dependencias	[IRQ-0005] Foro [OBJ-0007] Foro	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor encuentre inapropiado un comentario del foro.	
Precondición	El comentario debe existir	
Secuencia normal	Paso	Acción
	1	El actor Profesor (ACT-0001) borra un comentario del foro.
Postcondición	El comentario ha sido eliminado.	
Frecuencia esperada	1 vez por mes(es)	
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	

UC-0017	Modificación datos del profesor	
Versión	1.0	
Dependencias	[IRQ-0003] Administrador [OBJ-0014] Gestión de profesores	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee modificar alguno de los datos de los profesores.	
Precondición	El usuario debe existir.	
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0003) modifica los datos guardados del actor Profesor (ACT-0001) en la aplicación.
Postcondición	Los datos del usuario han quedado correctamente modificados.	
Frecuencia esperada	1 vez cada 3 meses.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

UC-0018	Borrar datos del profesor	
Versión	1.0	
Dependencias	[OBJ-0014] Gestión de profesores	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso se dé de baja a un profeso.	
Precondición	El usuario existe en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema elimina cualquier dato del usuario que es dado de baja.
Postcondición	Todos los datos del usuario han sido eliminados del sistema.	
Frecuencia esperada	1 vez por curso.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	

5.3.3 Requisitos no funcionales

NFR-0001	Multiplataforma
Versión	1.0
Dependencias	Ninguno
Descripción	El sistema deberá ser totalmente funcional en Internet Explorer, Firefox y Google Chrome.
Importancia	Importante
Urgencia	Hay presión
Estado	Validado
Estabilidad	Alta

Capítulo 6

Especificación de diseño del software

En este capítulo se pasará de la parte de análisis, a la resolución de dicho análisis mediante la fase de diseño, donde se comenzará a concretar cada una de las soluciones a los problemas planteados durante la recopilación de requisitos software.

6.1 INTRODUCCIÓN

Esta fase es fundamental para una posterior implementación óptima, pues en el diseño se plantean cada una de las necesidades reales que se solicitan desde la parte de análisis. Esta parte se denominará, por tanto, Dominio de la Solución.

Los métodos utilizados para explicar el funcionamiento y la integración de las nuevas clases son los diagramas de clases y secuencia. Gracias a los diagramas de clase se explica la arquitectura de las nuevas características y su integración con el sistema y con los objetos existentes. Además se implementarán los diagramas necesarios para explicar los patrones de diseño que van a utilizarse.

Mediante los diagramas de secuencia se explican los pasos que ha de seguir el sistema junto con el usuario para hacer funcionar las nuevas características que van a implementarse.

6.2 **ÁMBITO DEL SOFTWARE**

Es necesario para realizar un diseño del sistema acorde a las necesidades, concretar los principales objetivos del proyecto, los requisitos y las restricciones que se han de tener en cuenta. Estas características se definen en este capítulo para definir la situación inicial de la que se parte para realizar el diseño del sistema.

Se puede encontrar una definición de los objetivos y requisitos software de forma extensa en el Capítulo 5 de la memoria. A continuación se muestra de forma resumida aquellos aspectos relevantes que son necesarios para una correcta planificación del código y de la programación.

- Gestión de alumnos.
- Gestión de profesores.
- Gestión de ejercicios.
- Historial de pasos dados por el alumno.
- Contabilidad de las calificaciones.
- Foro.

Para la realización de estos objetivos existe una limitación que ha de cumplirse según los requisitos definidos en el “Análisis de Requisitos”; esta está especificada en los requisitos no funcionales y es que la aplicación web debe ser totalmente funcional en Internet Explorer, Firefox y Google Chrome.

6.3 **DISEÑO**

A la hora de afrontar el diseño de Calcul@, partimos de la aproximación al mundo de los objetos previamente realizada en el análisis, que representan las situaciones en cada uno de los escenarios posibles que van a darse en el proyecto. A la hora de realizar el diseño, hay que decidir la arquitectura a emplear, y posteriormente, se pasará a la implementación y construcción.

El principal objetivo que se pretende alcanzar en esta fase es obtener una idea clara de la arquitectura en la que se encuentra dividido el sistema.

6.3.1 Diseño de datos

Hay que tener en cuenta la importancia del almacenamiento de la información dentro del proceso de desarrollo y posterior continuación.

Al diseñar la base de datos se ha tenido en cuenta que el volumen de datos que se maneja puede llegar a ser muy grande, aumentando el tamaño de la aplicación así como la velocidad de ejecución de la misma, lo que podría ocasionar una bajada de rendimiento.

Para implementarla, se ha utilizado el lenguaje SQL para consultas, PHP con XAjax para conexión con la base de datos y un servidor Apache.

Se realizó un modelo lógico que se mostrará a continuación en el que se expone un esquema de los datos del sistema, la estructura de tablas de información así como la descripción de los atributos.

Tenemos un total de 8 tablas que forman la base de datos, las cuales están relacionadas de la siguiente forma, tal y como muestra el siguiente diagrama de entidad-relación:

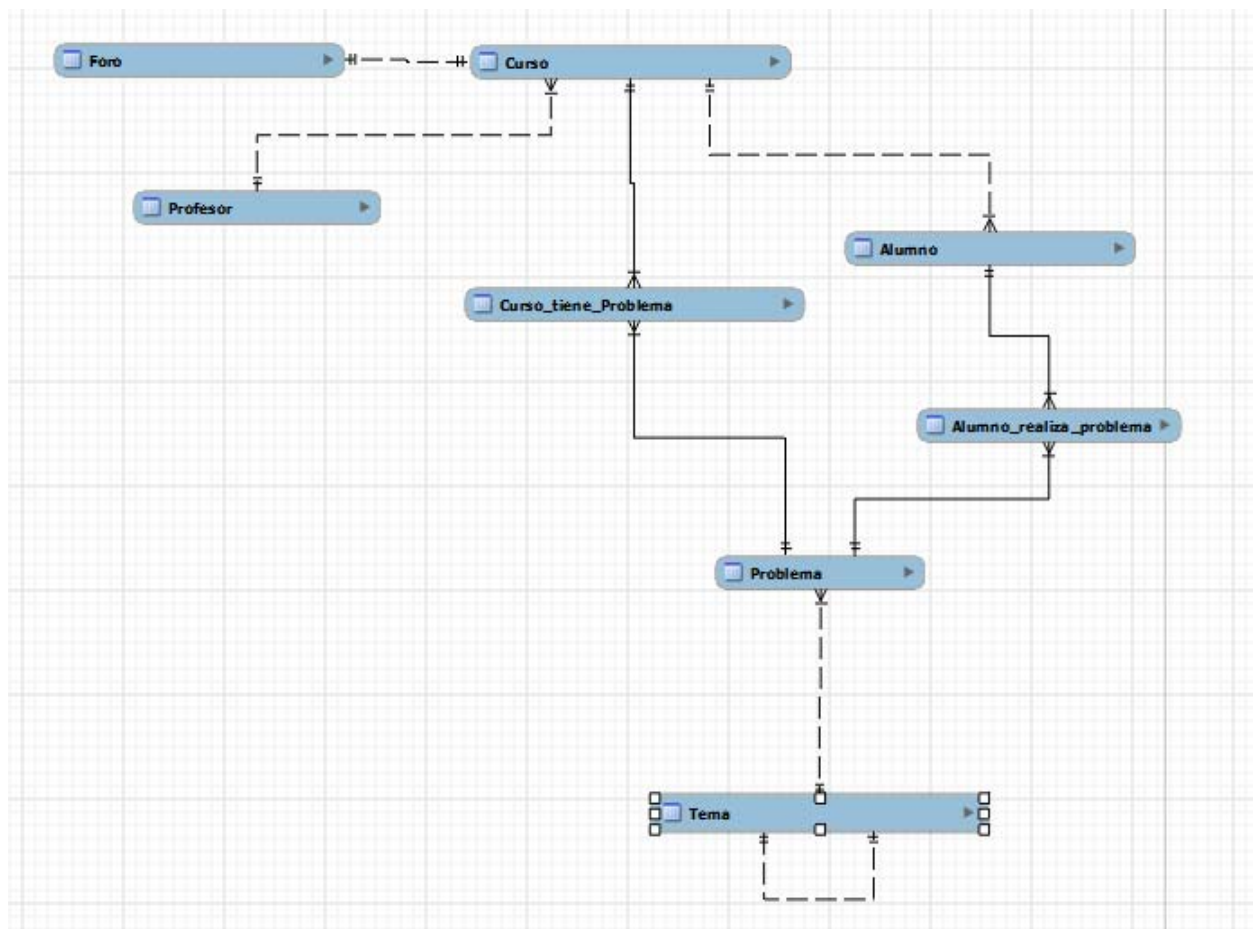


Ilustración 29: Estructura de la base de datos

A continuación se muestra una descripción más detallada de dicha base de datos, dentro de un catálogo donde se especifican los elementos y los atributos de los mismos que le dan sentido. Vamos a ir viendo cada una de las tablas.

6.3.1.1 Tabla alumno

Campo	Tipo	Nulo	Predeterminado
<u>idAlumno</u>	int(11)	No	None
nombre	varchar(20)	No	None
primer_apellido	varchar(45)	No	None
segundo_apellido	varchar(45)	No	None
telefono	varchar(9)	Sí	NULL
fecha_nacimiento	date	Sí	NULL
usuario	varchar(15)	No	None
password	varchar(15)	No	None
sexo	tinyint(1)	No	None
Curso_numero	int(11)	No	None
Curso_letra	char(1)	No	None

Tabla 1: Diccionario de datos. Tabla alumnos

6.3.1.2 Tabla alumno-realiza-problema

Campo	Tipo	Nulo	Predeterminado
<u>Problema-idProblema</u>	int(11)	No	None
<u>Alumno_idAlumno</u>	int(11)	No	None
pasos	varchar(255)	No	None
<u>fecha</u>	date	No	None
nota	float	Sí	NULL
queMandaAlumno	varchar(255)	Sí	NULL
texto	varchar(5000)	Sí	NULL

Tabla 2: Diccionario de datos. Tabla alumno_realiza_problema

6.3.1.3 Tabla curso

Campo	Tipo	Nulo	Predeterminado
<u>numero</u>	int(11)	No	None
<u>letra</u>	char(1)	No	None
Profesor_dni	int(11)	No	None

Tabla 3: Diccionario de datos. Tabla curso

6.3.1.4 **Tabla problema**

Campo	Tipo	Nulo	Predeterminado
<u>idProblema</u>	int(11)	No	None
nombre	varchar(45)	No	None
descripcion	varchar(45)	Sí	NULL
Tema_idTemas	int(11)	No	None
ayuda	varchar(255)	Sí	NULL

Tabla 4: Diccionario de datos. Tabla problema

6.3.1.5 **Tabla profesor**

Campo	Tipo	Nulo	Predeterminado
<u>dni</u>	int(11)	No	None
nombre	varchar(45)	No	None
primer_apellido	varchar(45)	No	None
segundo_apellido	varchar(45)	No	None
password	varchar(45)	No	None
Mail	Varchar(45)	No	None

Tabla 5: Diccionario de datos. Tabla profesor

6.3.1.6 **Tabla tema**

Campo	Tipo	Nulo	Predeterminado
<u>idTemas</u>	int(20)	No	None
parent_id	int(20)	No	None
position	int(20)	No	None
left	int(20)	No	None
right	int(20)	No	None
level	int(20)	No	None
title	text	Sí	NULL
type	varchar(255)	Sí	NULL

Tabla 6: Diccionario de datos. Tabla tema

6.3.1.7 **Tabla Curso-tiene-Problema**

Campo	Tipo	Nulo	Predeterminado
<u>Curso-numero</u>	int	No	None
<u>Curso-letra</u>	char	No	None
<u>Problema-idProblema</u>	int	No	None
enteros	bool	Sí	0
prioridad	varchar(45)	Sí	None
memoria	bool	Sí	0
tipoRepresentacionFraccion	varchar(45)	Sí	None
nDecimales	int(11)	Sí	None
agruparPalotes	int(11)	Sí	None
nParentesis	int(11)	Sí	None
parentesis	bool	Sí	0
palotes	bool	Sí	0
romanos	bool	Sí	0
teclasEstropeadas	varchar(45)	Sí	None
posTruncamiento	int(11)	Sí	None
redondeo	bool	Sí	0
reales	bool	Sí	0
fracciones	bool	Sí	0
decimales	bool	Sí	0
naturales	bool	Sí	0

Tabla 7: Diccionario de datos. Tabla Curso_tiene_Problema

6.3.1.8 **Tabla foro**

Campo	Tipo	Nulo	Predeterminado
<u>id</u>	tinyint(1)	No	0
autor	tinyint(1)	No	0
titulo	tinyint(1)	No	0
mensaje	tinyint(1)	No	0
fecha	tinyint(1)	No	0
respuestas	tinyint(1)	No	0
identificador	int(11)	No	None
Curso_numero	varchar(45)	No	None
Curso_letra	tinyint(1)	No	0

Tabla 8: Diccionario de datos. Tabla foro

6.3.2 Diseño de la interfaz

En el diseño de la interfaz se establece, principalmente, el aspecto visual de la aplicación y los aspectos relativos a la navegación.

El modelo elegido es diferente antes y después de autenticarse, a su vez siendo también diferente para profesores y alumnos.

6.3.2.1 Interfaz profesores

En el caso de los profesores, antes de autenticarse, en la pantalla podemos observar una parte principal con el formulario de entrada. En caso de haber olvidado la contraseña accederá a otra pantalla similar donde se piden los datos necesarios. Después de autenticarse, la interfaz se encuentra dividida en dos partes principales: el menú de la parte superior, desde donde podrán navegar hacia las demás opciones disponibles en su caso y la parte inferior, que es la zona principal de contenido, donde aparece la información según el profesor va navegando por la aplicación.



Ilustración 30: Ejemplo de interfaz antes de autenticarse los profesores



Ilustración 31: Ejemplo de interfaz después de autenticarse los profesores

6.3.2.2 Interfaz alumnos

En el caso de los alumnos, antes de autenticarse, en la pantalla podemos observar una parte principal con el formulario de entrada, al igual que en el caso de los profesores. Después de autenticarse, la interfaz cuenta con una imagen de fondo, que podrán elegir a través de la barra inferior y un menú desde donde podrán navegar hacia las demás opciones disponibles en su caso. Cuando seleccionen una de ellas, aparecerá una zona principal de contenido donde podrán acceder a la información que el profesor haya puesto a su disposición.



Ilustración 32: Ejemplo de interfaz antes de autenticarse los alumnos



Ilustración 33: Ejemplo de interfaz después de autenticarse los alumnos

6.3.2.3 Interfaz administrador

El administrador cuenta con una sencilla página desde la cual podrá realizar todas las acciones que tiene asignadas. En esta interfaz, sin ningún tipo de imagen de fondo, se pueden distinguir varias zonas con formularios desde donde realizarse el alta o baja de profesores, así como la modificación de sus datos o la posibilidad de enviar un email con el usuario y contraseña del profesor correspondiente. Además, en la parte inferior, cuenta con una tabla que muestra el listado de todos los profesores registrados en el sistema junto con sus datos.

Administrador de Calcul@

INFORMACIÓN GENERAL

Datos personales del profesor

Nombre:

Primer apellido:

Segundo apellido:

DNI:

Contraseña:

Email:

Actualizar

Nuevo profesor

DAR DE BAJA / ENVIAR EMAIL

Dar de baja

Usuario:

Contraseña:

Aceptar

Enviar email

Email:

Asunto:

Mensaje:

Enviar

¿Buscas un profesor? ¡Esta es la lista!

LISTA DE PROFESORES

Nombre	Apellidos	DNI	Contraseña	Email
Maria	Jimenez Ruiz	12554962	12345	maria-jr@gmail.com
Jose	Perez Martin	95685456	124578	jpm@hotmail.com
Nombre	Apellidos	DNI	Contraseña	Email

Ilustración 34: Ejemplo de interfaz administrador

Capítulo 7

Técnica de programación

Este capítulo nos servirá de guía para mostrar la implementación de nuestro sistema. Se especificarán los métodos y funcionalidades de las que se encarga cada módulo de implementación. Es un refuerzo para comprender mejor el código fuente de la aplicación de cara a futuros desarrolladores que necesiten nuestro sistema.

7.1 INTRODUCCIÓN

Como hemos visto, en la actualidad existen muchas aplicaciones de apoyo a la enseñanza. Cuando este proyecto todavía estaba ideándose, teníamos ante nosotros la posibilidad de estudiar uno ya existente y dotarlo de mejoras, o crear desde cero uno nuevo.

Nos decantamos por la última opción. Por lo tanto, desde que Calcul@ empezó a cobrar forma, pensamos en todo momento en que nuestro proyecto sería la base de otros en el futuro. Y en eso centramos nuestros esfuerzos: no queríamos numerosas pequeñas funcionalidades a medias, sino unos cimientos sólidos que cualquiera pudiera recuperar, entender, reutilizar y ampliar.

Programar código reutilizable y que no fuera, como sucede a veces, exclusivo para nuestro proyecto. Si no se tiene presente esto, a veces se podría acabar programando de forma que, cuando se cambie un detalle, ya no sirva el código tal cual está.

Como ya explicamos en el capítulo anterior, la parte gráfica del repositorio, sus colores representativos, tipos de letra y colores de ésta, son fácilmente modificables gracias a la utilización de la hoja de estilos CSS en la que están todos esos atributos claramente identificados por comentarios explicativos.

Para elegir ese aspecto y esos colores nos hemos basado en plantillas de HTML5 y después, las hemos cambiado casi en su totalidad para que ya no tuvieran un tamaño fijo con una única imagen de fondo, sino que se adaptasen a varios tamaños de ventana y diferentes gustos.

También atributos que no se refieren a los colores pero sí a la forma o colocación de algunos elementos que en un principio estaban en el código, pasaron a especificarse en la hoja de estilos.

Todas las rutas entre páginas son relativas. No hay rutas absolutas. De esa forma funcionará tanto si se ejecuta en la propia máquina de uno en un servidor local como si se sube a internet y se le confiere un dominio real desde el que acceder.

7.2 LENGUAJES UTILIZADOS

Durante el desarrollo de esta aplicación, hemos necesitado diversos lenguajes de programación para lograr completar todas sus funcionalidades, a la vez que conseguíamos un diseño dinámico e interesante.

El principal lenguaje utilizado en este proyecto es PHP, que se distingue por ser un lenguaje de programación de uso general de código del lado del servidor diseñado para el desarrollo web de contenido dinámico. Además, resulta muy útil la posibilidad de incorporarlo directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Asimismo, PHP ha evolucionado incluyendo una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Optamos por su utilización al poder ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Para la parte de diseño de la interfaz la opción más interesante era el uso de HTML 5, quinta revisión del lenguaje HTML. Aunque todavía se encuentra en modo experimental, es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. Este lenguaje hace las páginas compatibles con todos los navegadores web, incluyendo los de los teléfonos móviles y otros dispositivos modernos usados en la actualidad para navegar en Internet. Además es más ligero al ser más sencillo y simple el código, lo que permite que las páginas escritas en este lenguaje carguen más rápido en el navegador. También introduce infinidad de opciones que hasta ahora estaban vetadas a las páginas web realizadas con HTML, como insertar directamente un video, música y casi cualquier elemento.

Todo esto acompañado de JQuery y JavaScript, para simplificar la manera de interactuar con los documentos HTML, acompañado de la técnica de desarrollo web AJAX para crear aplicaciones interactivas donde es posible realizar cambios en las páginas sin necesidad de recargarlas.

7.3 IMPLEMENTACIÓN

7.2.1 Interfaces

Para el desarrollo de las interfaces de nuestro proyecto, tanto para la parte de alumnos como la parte de profesores, elegimos plantillas en HTML5, sobre las cuales realizamos las modificaciones precisas hasta llegar al resultado que esperábamos.

En este punto, nuestro objetivo era desarrollar una aplicación útil y fácilmente manejable sin olvidar la parte gráfica. De esta manera, queríamos incentivar la utilización de la aplicación por parte de los alumnos manejando colores llamativos e imágenes de fondo. En la parte baja de la pantalla, se muestra una colección de imágenes donde se da la posibilidad de elegir una de ellas y ponerla en pantalla completa si se desea.

Además, cabe destacar que el diseño de estilos está realizado en un fichero CSS para su mejor mantenimiento y evolución del código. Internamente podemos dividir la página principal en varios bloques, para esto nos hemos ayudado de la etiqueta `<div>` asociándole a cada una de ellas un identificador que la distinga dentro del código CSS, en concreto del fichero `'style.css'`. De esta manera, en caso de querer realizar alguna modificación futura sobre el estilo de la página será fácil e intuitiva llegar hasta él y poder realizar el cambio sobre el atributo deseado.

7.2.2 Bases de datos

Ya que consideramos este apartado uno de los puntos fundamentales para el desarrollo del proyecto y su futura utilización, se ha prestado especial atención sobre él.

Podemos distinguir dos maneras de interactuar con la base de datos. En primer lugar, en los casos donde es necesario recargar la página para poder visualizar la información introducida en la base de datos en ese momento utilizamos XAJAX. Mientras que si solamente deseamos su posterior utilización desde otra página utilizamos variables y funciones JavaScript.

Todas las funciones mencionadas anteriormente, así como las necesarias para la conexión y desconexión de la propia base de datos, se realizan desde el fichero `'matematicas_BBDD.php'`. Esto nos facilita su mantenimiento posterior y su bajo impacto en el caso de que queramos hacer una modificación en la base de datos.

Todas las acciones que se realizan contra la base de datos se han desarrollado en lenguaje SQL. Esta elección se debe a que nos permite especificar diversos tipos de operaciones (tales como consultas, inserciones, actualizaciones o eliminaciones) de una forma sencilla. Ya que una parte imprescindible del proyecto es el tráfico de información entre el servidor y la base de datos, hemos puesto especial atención en las pruebas que implicaban estas acciones. Comprobando cada una de ellas en el servidor utilizado antes de insertarlas en

el código del proyecto, de esta forma nos asegurábamos que de los fallos encontrados no surgieran de estas acciones y facilitando la posterior depuración del código.

7.2.2 Puntos clave del desarrollo

Dentro de esta aplicación, podemos diferenciar claramente tres partes: alumnos, profesores y administrador, no sólo en sus usos sino también en la manera de distribuir los ficheros que se han desarrollado.

Una de las principales consecuencias de esto es que, para su utilización, se dispondrán de tres enlaces de acceso diferentes. De esta manera, dificultamos que un alumno pueda adquirir el rol de profesor o un profesor maneje el rol de administrador, haciendo un uso indebido de la aplicación.

Por ello vamos a encontrarnos con tres archivos '*index.php*', cada uno en el paquete correspondiente a su parte de la aplicación, donde se encuentran las funciones que crean el entorno de las páginas web que se pueden visitar sin estar registrado en el sistema. Para el caso de alumno y profesor es el formulario de acceso a la aplicación, aunque en el caso de los profesores también podemos acceder a un formulario de recuperación de contraseña. En el caso del administrador, se accede a la página desde la cual puede realizar todas sus funciones permitidas.

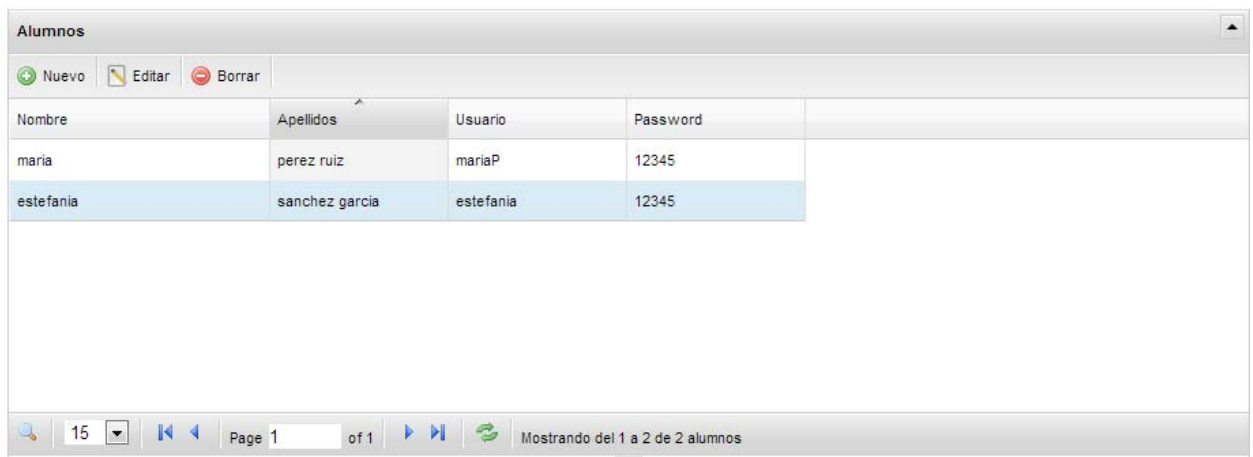
Si estamos en el caso de que el usuario rellena uno de los formularios e introduce datos erróneos, faltantes o inconsistentes podría ocurrir que el servidor los rechazara devolviéndolo al formulario de ingreso, entonces debería comenzar todo de nuevo. Para evitar esta situación, que puede llegar a ser frustrante para el usuario en caso de tener que rellenar muchos datos, utilizamos JavaScript en las validaciones. De esta manera, el usuario solo tendrá que modificar y/o completar el campo erróneo o faltante. Además, mostramos un breve texto de ayuda indicando cuál es la manera correcta de completar dicho campo. Estas validaciones incluyen la prohibición de que existan dos usuarios repetidos.

Una gran ventaja que nos hemos encontrado con la utilización de PHP es el control de sesiones. Una sesión, dentro de este contexto, es un espacio de almacenamiento de información en forma de variables que son persistentes al cambio entre la ejecución de un script y otro, y a su vez, permiten que cada espacio de almacenamiento sea único para cada visitante del sitio web de forma tal que dos usuarios distintos accediendo a la misma página no se interfieran entre sí. Esto quiere decir que los datos persisten para un usuario durante toda su permanencia en el sitio permitiéndole al servidor trasladar datos de una ejecución a otra para ese usuario.

Este mecanismo es sencillo de manejar en PHP y bastante transparente en cuanto a su implementación, ya que PHP tiene una función estándar para indicar que se usará una sesión, la cual es `session_start()`. Esta función es la que primero debe ejecutarse antes de enviar cualquier otra cosa al navegador y una vez ejecutada ya podemos acceder a cualquier variable

de sesión. Estas variables están contenidas en un array cuyo identificador es `$_SESSION[]`. Es imprescindible conocer esta parte para comprender el desarrollo de la aplicación, ya que durante toda la implementación hacemos uso de ello para poder saber qué datos debemos mostrar a cada uno de los alumnos registrados, utilizando el identificador de alumno que nos proporciona la variable de sesión.

Dentro de la implementación de nuestro proyecto, cabe destacar la existencia de dos tablas dinámicas creadas con *'plugins'* de JQuery. En ellas, a través de diferentes consultas a la base de datos extraeremos la información necesaria para completarlas. A continuación se muestra un ejemplo para la muestra de los alumnos matriculados en un curso determinado, contando con otra para visualizar, modificar o insertar problemas desde el apartado de profesores.



Nombre	Apellidos	Usuario	Password
maria	perez ruiz	mariaP	12345
estefania	sanchez garcia	estefania	12345

Ilustración 35: Tabla dinámica JQuery

Capítulo 8

Funcionalidades de la aplicación

En este capítulo se recogen las funcionalidades principales de la aplicación dependiendo de qué tipo de usuario esté haciendo uso de la misma: profesor, alumno o administrador. Puede servir de guía para el usuario del sistema, ya que aquí se detallarán los conceptos necesarios para el uso de la aplicación con imágenes de ejemplo y especificaciones de cada funcionalidad.

8.1 FUNCIONALIDAD PARA PROFESORES

En esta parte describiremos los pasos necesarios para el aprendizaje y manejo de la aplicación Web en la parte de los profesores, todo ello realizado mediante breves y concretas explicaciones y acompañado de imágenes que guiarán al usuario para la correcta utilización de la misma.

8.1.1 Antes de iniciar sesión

A continuación pondremos las funciones que se podrán realizar antes de iniciar sesión en la aplicación Web.

La situación que nos encontramos nada más acceder a la aplicación es la siguiente:



Ilustración 36: Funcionalidades profesores. Página de inicio.

En la parte superior de la pantalla encontramos un formulario de acceso a la aplicación, donde el usuario introducirá sus datos de registro para acceder a la parte principal de la aplicación. También hay un link por si se olvida la contraseña.

En el pie de página se encuentra la siguiente frase del célebre autor Miguel de Guzmán:

¿Dónde acaba el juego y dónde empieza la matemática seria?

Para muchos, la matemática, mortalmente aburrida, no tiene nada que ver con el juego.

En cambio, para la mayoría de los matemáticos, nunca deja de ser un juego, aunque, además, pueda ser muchas otras cosas.

Para nosotros esta frase representa en su totalidad el espíritu de Calcul@, un lugar donde las matemáticas nunca dejen de ser un juego a la vez que el niño aprende y se divierte con ellas.

Si pulsamos sobre el link para recordar la contraseña, avanzaremos a una nueva página web donde encontraremos un sistema de recordatorio de contraseña que únicamente se la reenviará al usuario si, además del DNI, introduce una dirección de correo electrónico y ésta se corresponde a la dirección asociada a ese usuario en el momento del registro. De esta forma evitamos que alguien haga un uso deshonesto del recordatorio de contraseña introduciendo números de DNI al azar o que conozca.

Cuando el profesor pulsa el botón Enviar, el administrador recibe un mensaje solicitando el recordatorio de la contraseña y, si todos los datos son correctos, se procede al envío de su contraseña a través de un email que les llegará automáticamente desde una dirección de correo creada expresamente para Calcul@: no.reply.calcula@gmail.com.



¿No puedes iniciar sesión?

Para restablecer tu contraseña, introduce tu DNI y la dirección de correo electrónico que hayas asociado a tu cuenta.

D.N.I:

Email:

Illustration: A 3D white figure stands next to a large yellow question mark, appearing to be in deep thought.

Ilustración 37: Funcionalidades profesores. Recordatorio contraseña.

8.1.2 Inicio de sesión

El registro realizado da garantías de seguridad obligando a los usuarios registrados a acceder introduciendo usuario, que en nuestro caso es el DNI sin letra, y contraseña que se contrastan en una base de datos y manteniendo sesiones activas mientras estos acceden; de forma que no pueda colarse ningún intruso no registrado ni mostrarse a un usuario el área personal de otro.



Ilustración 38: Funcionalidades profesores. Acceso profesores.

8.1.3 Una vez iniciada la sesión

Una vez que el usuario ha iniciado sesión en la aplicación se encontrará en la página principal de la aplicación.



Ilustración 39: Funcionalidades profesores. Inicio registrado.

El marco de la aplicación, dentro del usuario profesor, será siempre el mismo. Arriba, una barra de menú con las diferentes opciones de navegación y la imagen principal.

En el espacio central, se colocará el contenido correspondiente a la situación que se encuentre el usuario en cada momento. Al entrar, el usuario encontrará un pequeño texto con información sobre Calcul@, la plataforma virtual destinada a la enseñanza de matemáticas que compone la aplicación. A la derecha de este texto, el usuario podrá ver las últimas noticias, compuestas por los últimos movimientos realizados en la aplicación por los integrantes de su curso.

Durante este capítulo se seguirá el orden del menú superior de la aplicación.

8.1.3.1 Cursos

En esta sección de la aplicación se muestran diferentes opciones para tratar los cursos matriculados del usuario activo.

En la parte central derecha tenemos dos opciones: Matricular nuevos cursos, donde podemos matricular uno a uno nuevos cursos; y eliminar cursos existentes, para borrar todos los cursos matriculados del usuario activo.

En la parte central izquierda podemos ver una lista con todos los cursos existentes para dicho usuario. Ahora el usuario, al situarse sobre uno de los cursos, puede decidir si eliminar solo ese curso, simplemente, pulsando el botón correspondiente de la lista que aparece. Además puede dirigirse, de una manera directa, al listado de los alumnos matriculados o a la relación de los problemas disponibles para ese curso.

Además, en la parte derecha de la pantalla, contamos con un Calendario.

INICIO | CURSOS | PROBLEMAS | ALUMNOS | FORO | MANUAL

No hay problema imposible, aquí los resolvemos

CURSOS MATRICULADOS

Aquí podrá matricular nuevos cursos, así como consultar los cursos que ya tiene disponibles y acceder a las características de cada uno de ellos. También se podrán eliminar los cursos ya existentes.

Mis cursos: Opciones:

2-A > 4-A > Eliminar Alumnos Problemas

Matricular nuevos cursos Eliminar cursos existentes

CALENDARIO

May 2013

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Ilustración 40: Funcionalidades profesores. Cursos.

8.1.3.2 Problemas

En la siguiente pestaña tenemos las diferentes opciones posibles para trabajar con los problemas que el profesor quiere introducir a la aplicación con el fin de que sus alumnos los completen.

En primer lugar, siempre y cuando el profesor ya tenga seleccionado uno de los cursos disponibles en la pestaña “Cursos”, deberá seleccionar el modo de trabajo con los problemas. Se encuentran dos opciones: modo creación y modo asociación.

Para ambos modos se cuenta con un cuadro donde aparece la lista de todos los temas ya existentes en nuestra base de datos. Para cada uno de los temas, se dispone de diferentes apartados los cuales se anidan en forma de árbol.

Cuando el modo seleccionado es el de creación, si no hay temas disponibles o los problemas que se desean introducir no corresponden a los temas ya existentes en nuestra base de datos, tenemos la posibilidad de crear un tema nuevo utilizando los botones situados de la parte superior, de igual manera podremos crear nuevos apartados o subapartados. Además se dispone de diferentes opciones para editar los temas (por ejemplo, renombrar el título de un tema, cortar, pegar o borrar) o realizar una búsqueda entre los temas ya existentes. Hay que destacar que, hasta este punto, los problemas se generan de forma independiente al curso y será más adelante con el modo asociación cuando se pueda incluir dentro de él.

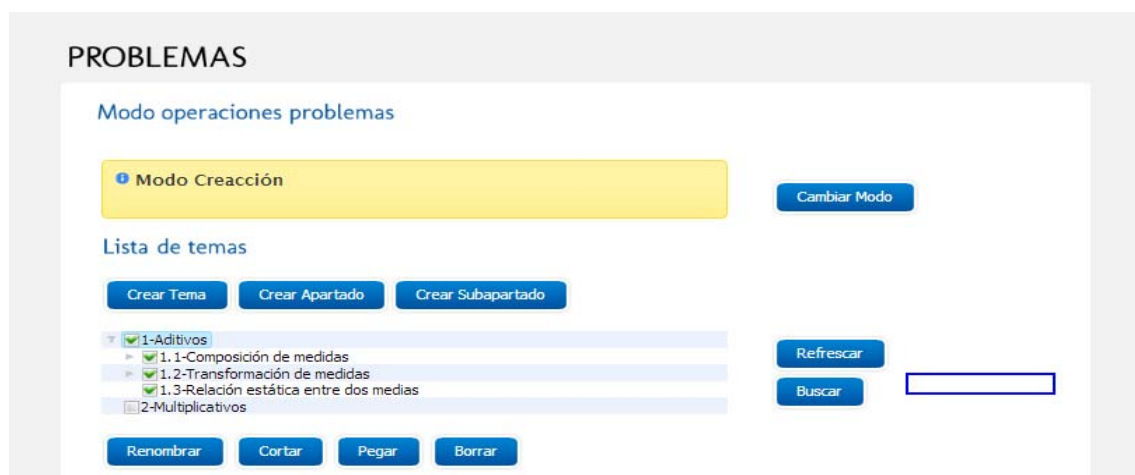
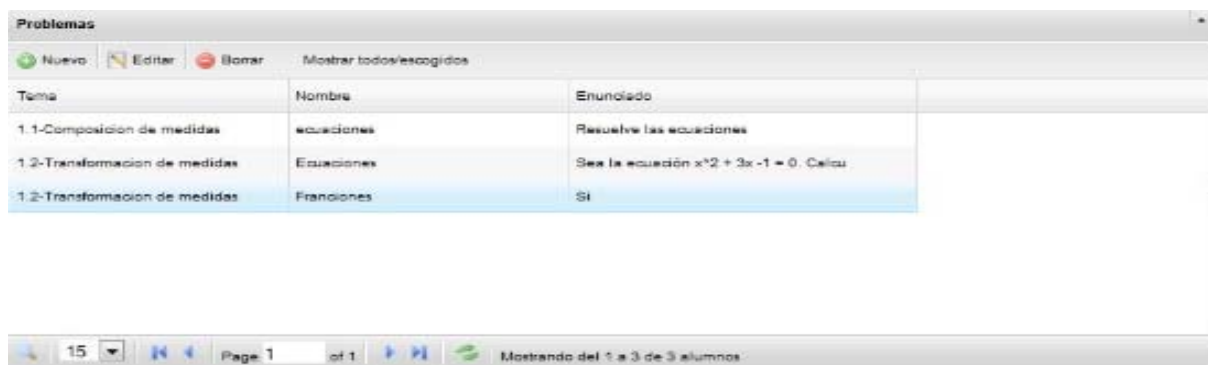


Ilustración 41: Funcionalidades profesores. Problemas (1).

Se pueden seleccionar uno o varios de los temas disponibles y con ello aparecerán, en la tabla situada inmediatamente debajo, todos los problemas que se hayan introducido anteriormente para dicho tema. Desde este momento tendremos la posibilidad de seleccionar uno de los problemas ya existentes o crear uno nuevo.

A continuación se puede ver en la pantalla una tabla dinámica, creada como explicamos en el Capítulo 8, donde se muestra el tema, el nombre y el enunciado de los problemas ya insertados para el tema seleccionados por el profesor.



Tema	Nombre	Enunciado
1.1-Composicion de medidas	ecuaciones	Resuelve las ecuaciones
1.2-Transformacion de medidas	Ecuaciones	Sea la ecuación $x^2 + 3x - 1 = 0$. Calcu
1.2-Transformacion de medidas	Fracciones	Si

Ilustración 42: Funcionalidades profesores. Problemas (2).

En caso de que se quiera insertar un nuevo problema, debemos pulsar el botón 'Nuevo' situado en la parte superior izquierda de la tabla. Al elegir esta acción, se deberá completar un formulario con los datos del problema para poder crearlo en la base de datos.

De la misma manera funciona el botón 'Borrar', con él podemos suprimir uno de los problemas de la lista. Si se pulsa dicho botón y aceptamos el mensaje de confirmación de eliminación del problema, debemos tener en cuenta que el problema se borrará de la base de datos de manera permanente y la única opción de recuperarlo será insertarlo de nuevo.

Para realizar la selección de problemas de un curso y asociarles la calculadora oportuna, se debe elegir el modo asociación pulsando el botón "Cambiar modo". Desde esta opción, se tendrá la posibilidad de asociar uno o varios problemas de los ya creados anteriormente a uno de los cursos disponibles para el usuario actual.

El primer paso a realizar es seleccionar el problema deseado dentro de la lista que aparece en la tabla y pulsar el botón 'Editar' que aparece en la parte superior. De esa manera, nos aparecerá un cuadro similar al del caso anterior donde se debe elegir la calculadora que se prefiere para dicho problema.

En primer lugar, contamos con todos los botones disponibles en la calculadora, si pulsamos alguno de ellos este botón quedará inhabilitado en la calculadora que el alumno pueda utilizar para resolver este problema. Además realizando la selección oportuna del formulario que se encuentra situado justamente debajo de dichos botones podremos elegir entre el dominio numérico con el que el alumno debe trabajar, el número de decimales, la prioridad de los operadores, si se desea realizar redondeo o truncamiento, si tiene que haber o no memoria, si se deben evaluar las expresiones complejas por partes o la opción de tener calculadoras de números romanos o de fracciones. Se debe completar todos los campos para que el registro del nuevo problema se complete con éxito.

Ilustración 43: Funcionalidades profesores. Crear problema.

Es imprescindible la realización de los pasos que se acaban de detallar si se quiere realizar de una manera correcta el proceso de inserción o selección de problemas.

A continuación se muestra una vista general de la pestaña que se acaba de explicar:

Ilustración 44: Funcionalidades profesores. Problemas(3).

8.1.3.3 Alumnos

Dentro de esta pestaña se nos muestran las diferentes posibilidades que un profesor tiene para tratar a los alumnos matriculados dentro un curso determinado.

A continuación se muestra el aspecto de dicha pestaña:



Nombre	Apellido	Usuario	Password	Nota media
Susana	Flores Aviles	Susana	45678	3
Sara	Jimenez Hernandez	SaraJ	12345	7
Pedro	Trujillo Perez	Pedro	Pedro	5

Porcentaje aprobados

50% Aprobados
50% Suspenzados

Ilustración 45: Funcionalidades profesores. Alumnos.

Al igual que en el caso de los problemas, en primer lugar, se debe seleccionar uno de los cursos disponibles para el profesor que ha iniciado la sesión. Esto se realiza en el cuadro de selección disponible en la parte superior de la zona principal de la página actual. Al realizar esta acción se actualizará la tabla que muestra todos los alumnos y los datos más relevantes de

ellos matriculados en el curso elegido, como son el nombre y apellidos, usuario y password y la nota media de todos los problemas que ha resuelto hasta el momento dicho alumno.

Los botones disponibles en esta tabla son parecidos al caso de los problemas explicados en el apartado anterior, simplemente con la diferencia de que en la ventana emergente se solicitan los datos correspondientes a los alumnos.

Un detalle interesante a destacar de esta tabla es la posibilidad de actualizarla al momento, así como la elección del número de elementos que se quieren visualizar en cada una de las páginas de la tabla.

Además, en la parte inferior de la pantalla se dispone de un gráfico que muestra el porcentaje de aprobados y suspensos, calculado a partir de las notas medias de todos los alumnos pertenecientes al curso seleccionado.

Desde la tabla, el profesor también tiene la posibilidad de calificar los problemas que el alumno haya resuelto hasta ese momento pulsando el botón “Nota”. Al realizar esta opción se nos dirige a otra pantalla similar que consta de una tabla donde aparecen todos los problemas que ha resuelto el alumno. Cuando se selecciona uno de ellos, en primer lugar al pulsar el botón “Detalles” se completan los cuadros inferiores donde se muestra el enunciado del problema, los pasos realizados para resolverlo y la solución que el alumno haya decidido enviarle a su profesor. Una vez estudiados todos estos datos, el profesor ya puede calificar al alumno desde el botón “Editar nota”

ALUMNO

Sara Jimenez Hernandez

Ejercicios resueltos

Detalles

Editar Nota

Tema	Nombre	Fecha	Nota
1.1.1-Pregunta por el todo	Suma total	2013-08-30	7
1.1.1-Pregunta por el todo	Sumas	2013-08-29	6
1.1.1-Pregunta por el todo	Restas	2013-08-28	8

15 of 1 Page 1 of 1 Mostrando del 1 a 3 de 3 alumnos

Enunciado

Pasos realizados

10+14=24

Comentarios adicionales

Ilustración 46: Funcionalidades profesores. Calificación de ejercicios.

8.1.3.4 **Foro**

El foro es un elemento compartido entre alumnos y profesores. En el caso de los profesores, se debe elegir uno de los cursos disponibles ya que únicamente se puede ver el foro de un curso en concreto, no de varios cursos a la vez.

A continuación se mostrará un listado de los temas abiertos, donde se dará la posibilidad de ver uno de estos temas o de iniciar uno nuevo.

En caso de que se quiera iniciar un nuevo tema, simplemente deberemos pulsar el enlace correspondiente. Después de esto, deberemos completar los campos '*Título*' y '*Mensaje*' para que se actualice en la base de datos el foro correspondiente al curso seleccionado.

Si solamente se quiere leer uno de los mensajes ya publicados, se debe pulsar el enlace '*Ver*' situado a la izquierda del título del mensaje. En la misma pantalla que vemos el mensaje elegido, se tendrá la posibilidad de responder a dicho mensaje.

8.1.3.4 **Calificar**

Desde esta pestaña se realizan las mismas funciones que al seleccionar uno de los alumnos del curso correspondiente y pulsar el botón "Nota". Es un acceso rápido a esta función ya explicada en el apartado "8.1.3.3 Alumnos".

8.1.3.5 **Manual**

La única función de la pestaña '*Manual*' es abrir un archivo PDF donde se recogen las principales indicaciones que deben seguir los profesores que quieran hacer uso de esta aplicación.

8.2 **FUNCIONALIDAD PARA ALUMNOS**

Dentro de este apartado detallaremos los pasos necesarios para el aprendizaje y manejo de la aplicación Web en la parte de los alumnos, para facilitar la tarea se proporcionarán breves explicaciones, acompañadas de imágenes, que guiarán al usuario para la correcta utilización de la misma.

8.2.1 Antes de iniciar sesión

El sistema actual no cuenta con funciones disponibles para los alumnos antes de registrarse en la aplicación, por tanto el primer paso que se debe dar si se quiere hacer uso de ella es el registro.

8.2.2 Inicio de sesión

La situación que nos encontramos nada más acceder a la aplicación es la siguiente, donde los alumnos deberán completar su usuario y contraseña:



Ilustración 47: Funcionalidades alumnos. Acceso.

En caso de que alguno de los datos introducidos por el niño no sean correctos, se mostrará un mensaje indicando que el usuario no se encuentra en la base de datos o la contraseña es incorrecta.

Si el alumno no está registrado o ha olvidado su contraseña, no tendrá posibilidades de hacerlo por sí mismo sino que deberá ponerse en contacto con su profesor.

8.2.3 Una vez iniciada la sesión

Una vez que el alumno ha iniciado sesión se encontrará en la página principal de la aplicación, donde podrá elegir el fondo de pantalla que más le guste.



Ilustración 48: Funcionalidades alumnos. Principal.

Además cuenta de un menú en la parte superior de la pantalla, con las opciones: '¿Quién soy?', 'Problemas', 'Evaluación' y 'Foro'. A continuación, se detallará el funcionamiento de cada una de estas posibilidades.

8.2.3.1 ¿Quién soy?



Ilustración 49: Funcionalidades alumnos. ¿Quién soy?.

Dentro de este apartado el alumno tendrá la posibilidad de visualizar sus datos personales, tal y como están guardados en la base de datos. En caso de que alguno de ellos sea incorrecto, el estudiante deberá ponerse en contacto con su profesor lo antes posible para que sea éste el encargado de modificar los datos erróneos desde su parte de la aplicación como comentamos en el apartado '8.1.3.3 Alumnos'.



Ilustración 50: Funcionalidades alumnos. Datos personales.

8.2.3.2 Problemas



Ilustración 51: Funcionalidades alumnos. Problemas.

En esta pestaña, se publican una relación de temas con sus ejercicios correspondientes que han de ser resueltos por los alumnos. Para que estos problemas sean visibles para los estudiantes, los profesores han de habilitarlos previamente tal y como se especifica en el apartado '8.1.3.2 Problemas'.



Ilustración 52: Funcionalidades alumnos. Problemas disponibles.

Como se indica en la pantalla es recomendable realizar los ejercicios en el orden en que aparecen. Sin embargo, para distinguir más fácilmente los problemas que ya han sido resueltos por el estudiante registrado utilizamos un código de colores; este código consiste en que el color azul representa los ejercicios que todavía no han sido realizados mientras que el color morado indica que el alumno ya ha completado el ejercicio marcado.

Cuando un alumno quiere realizar un determinado ejercicio solamente tiene que pulsar sobre su nombre y se le trasladará a la página que se muestra a continuación.

En ella se puede observar una zona principal, en el centro de la pantalla, donde se encuentra el enunciado del problema a resolver. Justo debajo, se puede ver el cuadro de texto Solución donde el alumno debe solventar el problema y, si lo desea, escribir algún tipo de indicación que considere relevante e interesante para la evaluación del ejercicio.

A la derecha de este apartado, se puede encontrar un texto que muestra ayuda teórica y breves indicaciones de la manera de resolver el problema en cuestión, redactado por el profesor.

Esta pantalla dispone de dos botones: enviar y volver. Si el alumno decide que su solución es la correcta y quiere que sea evaluada por el profesor, debe pulsar el botón 'Enviar' y quedará registrada en la base de datos tanto la solución escrita por él como todos los pasos que haya dado con la calculadora para resolverlo. Desde ese momento, el profesor podrá visualizar en el perfil del alumno todos estos datos y calificarlo con la nota oportuna. Si por el contrario, desea volver a la página principal de la aplicación tan solo deberá pulsar el botón 'Volver'.



Ilustración 53: Funcionalidades alumnos. Resolución de problemas.

Como se indica al pie de página, en este caso, el alumno puede usar la calculadora de la aplicación. Por ello en la parte inferior de la pantalla se muestra la calculadora específica que el profesor considera oportuna para la resolución de este problema. Esta calculadora tiene una apariencia similar a la proporcionada por el sistema operativo Windows, pero con una estética infantil.

Dentro de esta aplicación disponemos de 6 tipos de calculadoras generales, que el profesor puede personalizar para cada uno de los problemas con la opción 'Teclas Estropeadas', es decir, teclas anuladas para el uso en problemas concretos, ya sean números u operadores. Las calculadoras de nuestra aplicación deberían soportar las siguientes opciones:

- En primer lugar, se tiene la posibilidad de elegir el dominio numérico en el que se quiere resolver el problema. Las opciones disponibles serán: naturales, enteros, decimales, fracciones/racionales y reales.
- En el caso de decimales, es posible que el profesor configure el número de cifras decimales a manejar y elegir el tipo de redondeo o truncamiento. Con esta

calculadora los alumnos podrán resolver las operaciones básicas (suma, resta, multiplicación y división), además podemos calcular tantos por ciento (%). También se permite la opción de trabajar con números negativos, utilizando la tecla de signo.

- También se permite ejecutar las expresiones complejas por partes, es decir, en casos con paréntesis nos da el resultado paso a paso permitiendo ver la evolución del cómputo. Además permite configurar los niveles de paréntesis anidados posibles, incluyendo la opción de no permitir paréntesis.

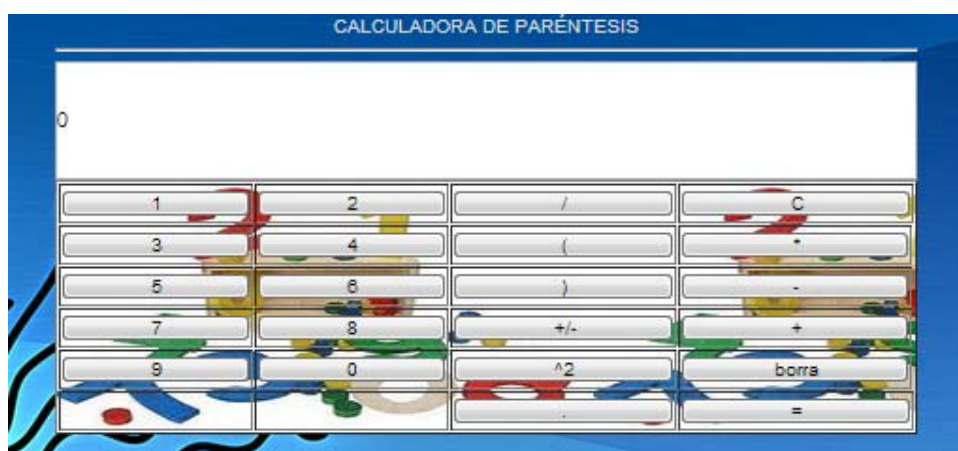


Ilustración 54: Funcionalidades alumnos. Calculadora.

- También incluimos la posibilidad de operar con números romanos, para naturales, con la siguiente calculadora. En ella se utilizan las normas establecidas para las operaciones y conversiones con este tipo de números.



Ilustración 55: Funcionalidades alumnos. Calculadora números romanos.

- Se dispone de una calculadora especial para manejar fracciones, donde el alumno debe escribir en orden las fracciones siguiendo el orden numerador y a continuación denominador de cada una de ellas.



Ilustración 56: Funcionalidades alumnos. Calculadora fracciones.

- Una característica común de todas las calculadoras anteriores es la opción de tener memoria o no, siendo el profesor el que elige si esta opción está disponible. También incluimos la opción de configurar la prioridad de operadores.

8.2.3.3 Evaluación



Ilustración 57: Funcionalidades alumnos. Evaluación.

Dentro de esta pestaña el alumno podrá consultar la calificación que el profesor ha otorgado a cada uno de los problemas que ha resuelto dicho alumno hasta el momento. La función de esta pestaña es meramente informativa.

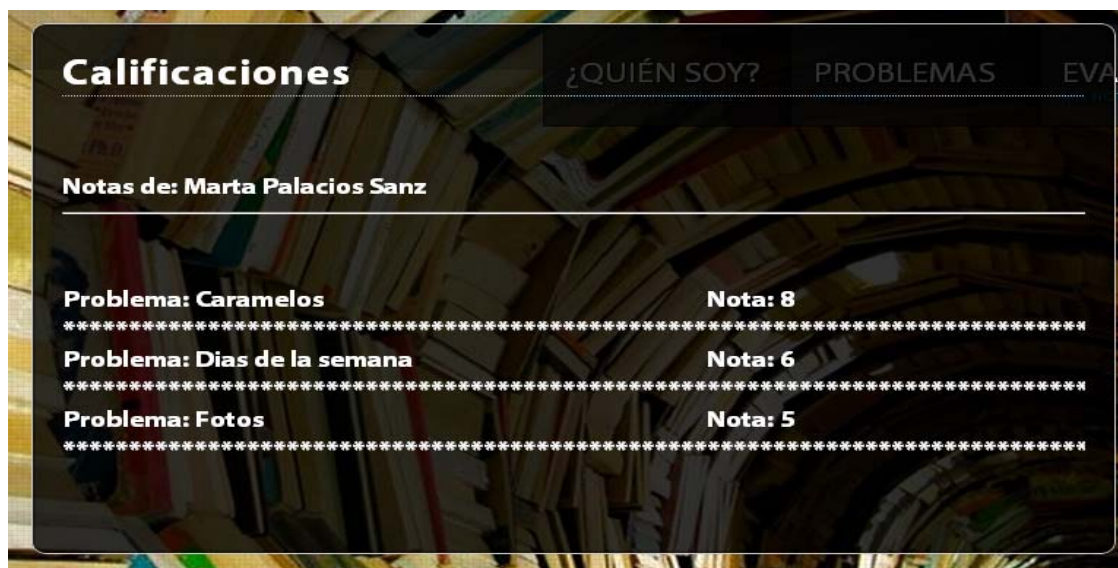


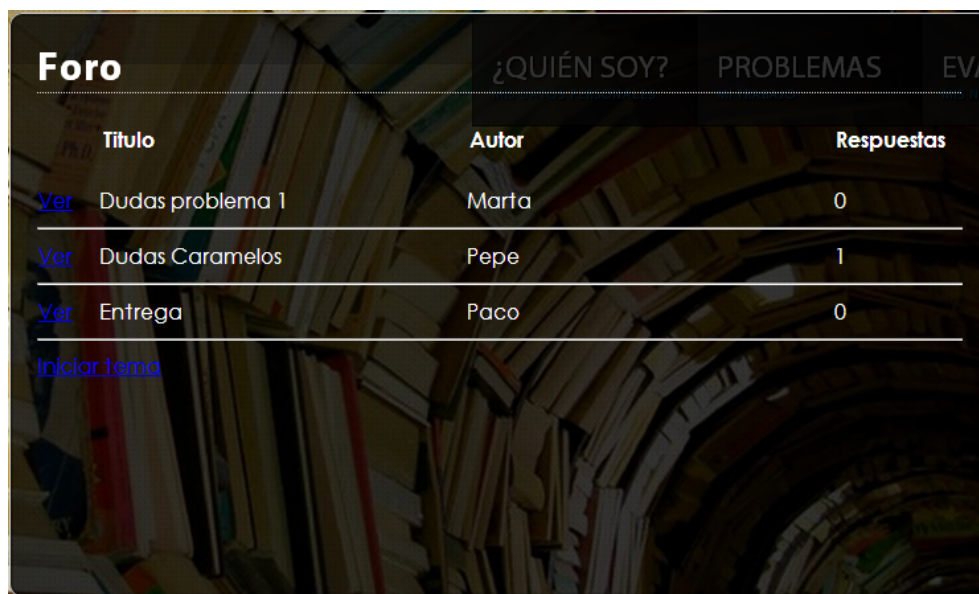
Ilustración 58: Funcionalidades alumnos. Calificaciones.

8.2.3.4 Foro



Ilustración 59: Funcionalidades alumnos. Foro.

Al pulsar sobre esta pestaña, aparece el listado de los temas abiertos en el foro correspondiente al curso en el que está matriculado el alumno actual. Desde este punto, el alumno podrá elegir entre iniciar un nuevo tema, para crear un hilo de debate nuevo, o ver uno de los temas existentes y responder si lo desea.

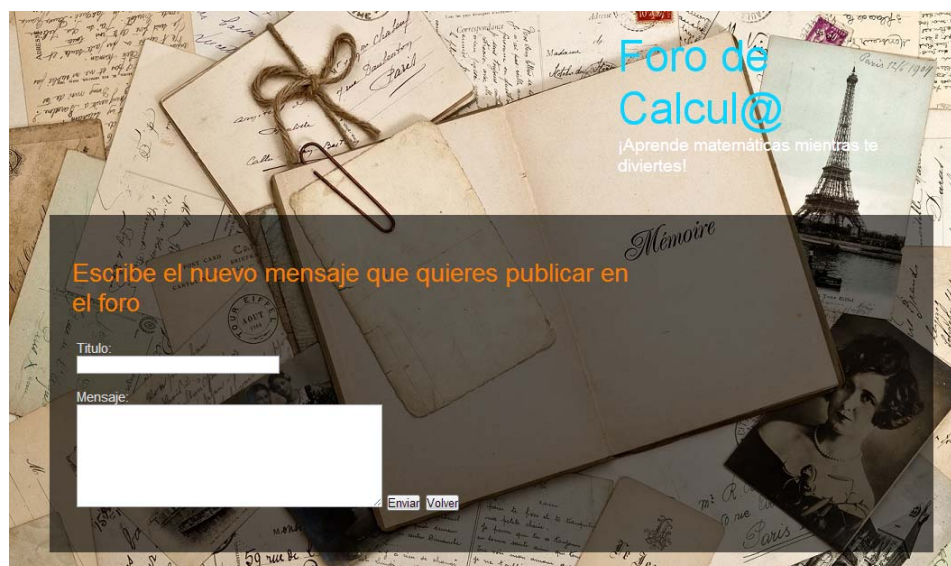


Titulo	Autor	Respuestas
Ver Dudas problema 1	Marta	0
Ver Dudas Caramelos	Pepe	1
Ver Entrega	Paco	0

[Iniciar tema](#)

Ilustración 60: Funcionalidades alumnos. Temas foro.

En caso de que el alumno elija la opción de Iniciar un nuevo tema, se dirigirá al siguiente formulario donde debe rellenar el título y el mensaje que desea publicar. El resto de datos, tales como la fecha, nombre de autor, identificador o número de respuestas no será necesario que los rellene el autor ya que serán guardados automáticamente por el sistema.



Foro de Calcul@
¡Aprende matemáticas mientras te diviertes!

Écris le nouveau message que tu veux publier en le forum

Titulo:

Mensaje:

Ilustración 61: Funcionalidades alumnos. Formulario foro.

Una vez completados todos los campos anteriores, si pulsamos el botón 'Enviar', nuestro mensaje se queda registrado en la base de datos y pasa a formar parte de la colección de temas de ese curso.

Si por el contrario queremos ver un mensaje ya publicado se debe pulsar sobre el enlace 'Ver' que nos trasladará a la página correspondiente a dicho mensaje donde se nos mostrarán también sus respuestas, si las hubiera.

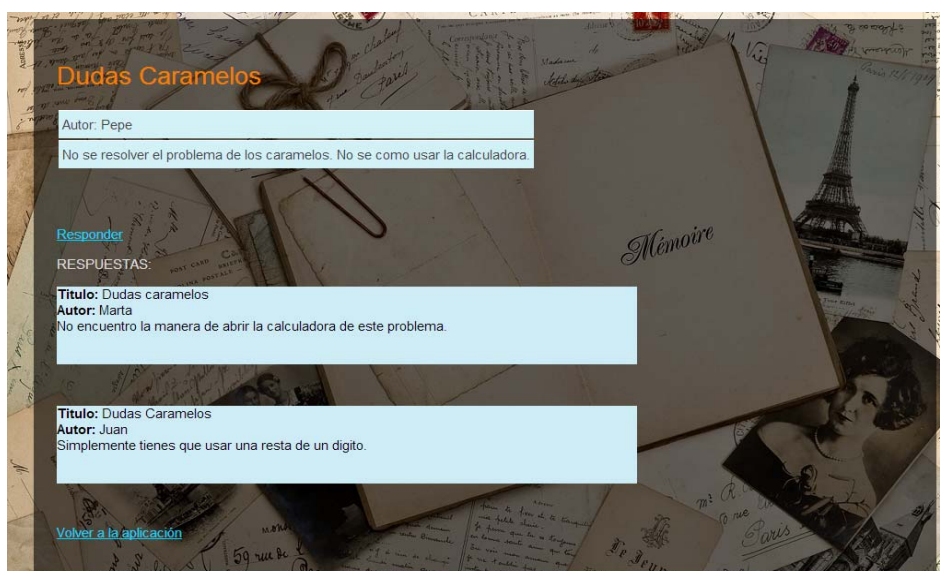


Ilustración 62: Funcionalidades alumnos. Respuestas foro.

Si se quiere responder en este hilo del foro, tan solo debemos pulsar el enlace 'Responder' desde donde nos dirigiremos al mismo formulario que en el caso de 'Iniciar tema'. Sin embargo, en esta ocasión de manera interna el sistema guardará el mensaje en la base de datos como respuesta a un hilo ya abierto, y no como comienzo.

8.3 FUNCIONALIDAD PARA ADMINISTRADOR

Dentro de este apartado detallaremos los pasos necesarios para el aprendizaje y manejo de la aplicación Web en la parte del administrador, para facilitar la tarea se proporcionarán breves explicaciones, acompañadas de imágenes, que guiarán al usuario para la correcta utilización de la misma.

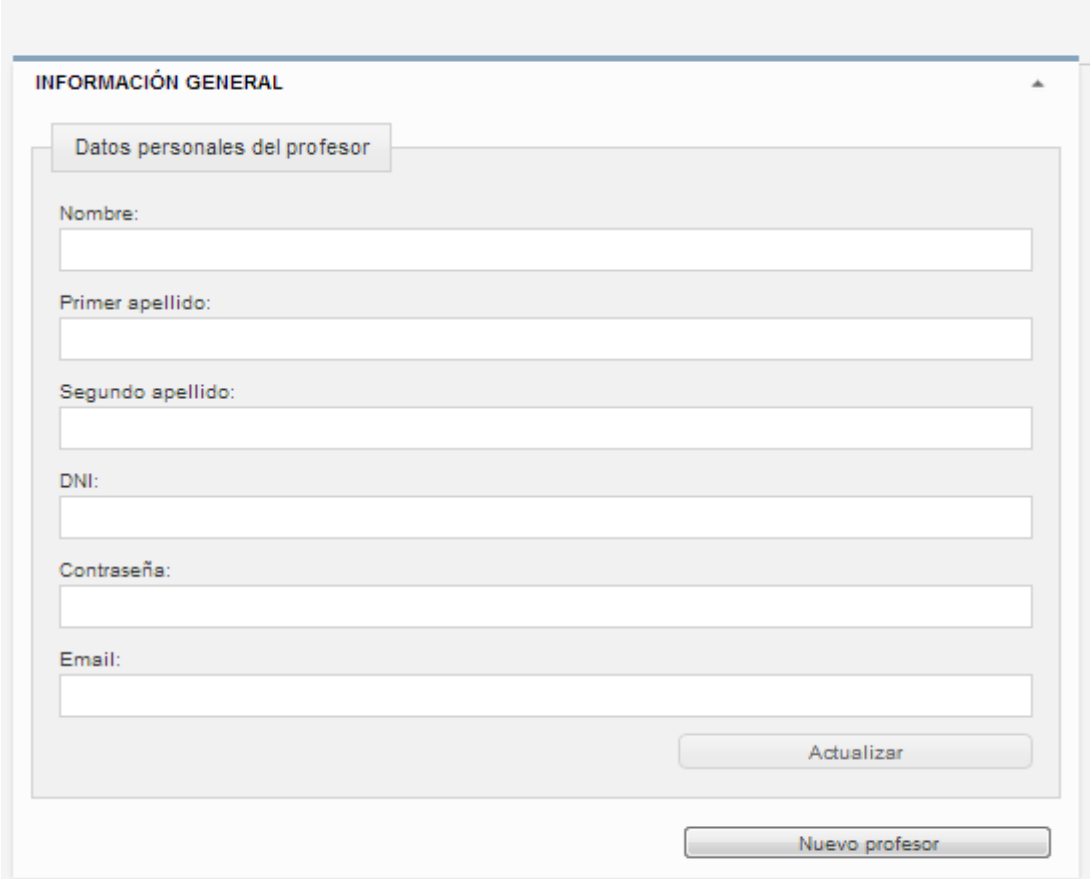
8.3.1 Antes de iniciar sesión

El sistema actual no cuenta con función de registro para el administrador, dado que se asume que éste es único y que ningún otro usuario tendrá la posibilidad de acceder a dicha página.

8.3.2 Dar de alta / Actualizar profesor

Se dispone de una misma zona dentro de la interfaz del administrador para realizar estas dos acciones. Para ambos casos, cuando el administrador desee registrar a un nuevo

usuario debe rellenar el siguiente formulario que solicita los siguientes datos personales: nombre, apellidos, DNI, contraseña y email del profesor. La única diferencia a la hora de hacer una u otra acción es el botón que se debe pulsar para realizarlas.



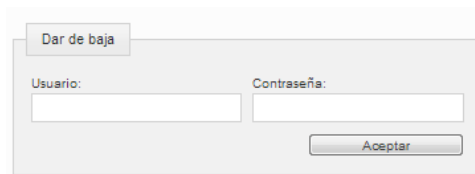
El formulario se titula "INFORMACIÓN GENERAL" y contiene un subencabezado "Datos personales del profesor". A continuación, hay campos de entrada para "Nombre:", "Primer apellido:", "Segundo apellido:", "DNI:", "Contraseña:" y "Email:". En la parte inferior derecha del formulario, hay un botón "Actualizar". Debajo del formulario, hay un botón "Nuevo profesor".

Ilustración 63: Funcionalidades administrador. Dar de alta/ Actualizar profesor.

Hemos desarrollado un sistema de validación instantáneo que comprueba la idoneidad de los campos al rellenarlos, en caso de no ser correctos la secuencia de caracteres introducidos en cada campo mostrará mensajes indicativos de los pasos a seguir para corregirlo. Ya sea porque el campo está vacío y debe ser rellenado o porque no cumple los requisitos (por ejemplo, el formato del DNI de 8 dígitos sin letra, un email válido, que ambos campos de contraseña coincidan...)

8.3.3 Dar de baja profesor

Para tramitar la baja de un profesor, el administrador simplemente deberá rellenar los campos de usuario y contraseña correspondientes. Esta acción borrará al profesor en cuestión de la base de datos.

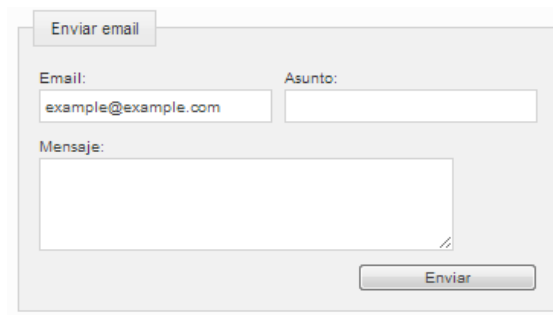


Formulario 'Dar de baja' con campos para Usuario y Contraseña, y un botón Aceptar.

Ilustración 64: Funcionalidades administrador. Dar de baja profesor.

8.3.4 Enviar email

El administrador utilizará esta sección para comunicarse con los profesores registrados en la aplicación mediante un sistema de correo.



Formulario 'Enviar email' con campos para Email (prellenado con example@example.com), Asunto, y un área de texto para el Mensaje. Incluye un botón Enviar.

Ilustración 65: Funcionalidades administrador. Enviar un email.

8.3.5 Listado de profesores

Para facilitar la tarea del administrador dispone de una tabla donde se muestran todos los profesores registrados en el sistema. De esta manera, se tiene una visión clara y un rápido acceso a todos los datos almacenados.

¿Buscas un profesor? ¡Esta es la lista!

LISTA DE PROFESORES				
Nombre	Apellidos	DNI	Contraseña	Email
Maria	Perez Ruiz	23485777	12345	maria.p@gmail.es
Javier	Heras Martin	51478477	12345	fjheras@gmail.com
Nombre	Apellidos	DNI	Contraseña	Email

Ilustración 66: Funcionalidades administrador. Listado de profesores.

Capítulo 9

Conclusiones y líneas de trabajo futuras

9.1. CONCLUSIONES

La conclusión más importante es la consecución de todos los objetivos que se marcaron en un principio y que ha conseguido llevar a cabo la idea que se tenía inicialmente de conseguir una herramienta de apoyo a la enseñanza de matemáticas en educación primaria viable.

Desde el punto de vista personal, los objetivos se han cumplido satisfactoriamente:

- **Adquisición de experiencia:** Mediante la realización del Proyecto de Fin de Carrera, se gana experiencia realizando un producto software que se asemeje a las aplicaciones profesionales del mercado. Esta experiencia será de gran utilidad en un futuro cuando se participe en otros proyectos, ya que durante el desarrollo han surgido un buen número de problemas que se han tenido que solventar de la manera más rápida y eficiente posible.
- Se han puesto en la **práctica conocimientos adquiridos** durante la carrera.
- **Gran ampliación de conocimientos:** Profundización en el aprendizaje del lenguaje PHP, en el manejo y configuración de servidores web y en el gestor de bases de datos MySQL. Además de en el paradigma de la orientación a objetos, utilizando sus metodologías y diseñando para la reutilización.
- Se ha conseguido dotar a la aplicación de un aspecto dinámico e intuitivo y, sobre todo, se ha conseguido crear una aplicación totalmente válida y funcional.

9.2 **LÍNEAS DE TRABAJO FUTURAS**

En este apartado de la memoria se describen las líneas de trabajo futuras que se podrían seguir en una posterior ampliación del producto desarrollado.

- **Nuevas formas de acceso:** Intentar que Calcul@ tenga operativa una página web para el acceso con dispositivos móviles, ya que el creciente aumento de dispositivos con acceso a Internet hacen que sea un punto a tener muy en cuenta.
- **Utilizar un servicio de alertas al móvil:** Este servicio mejoraría la comunicación entre los usuarios. Consiste en implementar un servicio para alertar e informar a los usuarios que así lo deseen. Por ejemplo, en el caso de los profesores podrían recibir alertas cuando deba corregir problemas resueltos por sus alumnos mientras que los alumnos podrían recibir avisos de los nuevos ejercicios que deben realizar.
- **Internacionalizar la página:** Un punto a mejorar es la posibilidad de hacer la página accesible en más idiomas.
- **Calendario - agenda:** El calendario disponible en la página de los profesores se podría convertir a su vez en una agenda, donde se pudieran anotar las tareas a realizar en los próximos días.

Bibliografía

Libros y documentos

- [1] Durán Toro, Amador y Bernárdez Jiménez, Beatriz. *“Metodología para la Elicitación de Requisitos de Sistemas Software”*. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. 2002.
- [2] Eguíluz Pérez, Javier. *“Introducción a JavaScript”*. 2009.
- [3] Schmitt, Christopher, Trammel, Mark, Marcotte, Ethan, Orchard, Dunstan, Sominey, Todd. *“CSS: Hojas de estilo en cascada para el diseño web”*. Anaya Multimedia. 2006.
- [4] Silberschatz, Abraham, Korth, Henry F., Sudarshan, S. *“Fundamentos de Bases de Datos”*. Universidad de Yale, Universidad de Lehigh, Instituto tecnológico indio. 2006.
- [5] Glass, Michael, Le Scouarnec, Yann, Naramora, Elizabeth, Mailer, Gary, Stolz, Jeremy, Gerner, Jason. *“Desarrollo web con PHP, Apache y MySQL”*. Anaya Multimedia. 2004.
- [6] Pressman, Roger S. *“Ingeniería del Software: un enfoque práctico”*. McGraw-Hill. 2010.

Internet

- [7] Fernando Posada, aplicaciones TIC para la enseñanza de las matemáticas en Primaria (<http://es.slideshare.net/fernandoposada/aplicaciones-tic-para-la-enseanza-de-las-matemticas-en-primaria>).
- [8] <http://es.wikipedia.org/wiki/Moodle>
- [9] http://es.wikipedia.org/wiki/Proceso_Unificado
- [10] https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos
- [11] <http://es.wikipedia.org/wiki/PHP>
- [12] <http://es.wikipedia.org/wiki/JavaScript>
- [13] <http://es.wikipedia.org/wiki/HTML>
- [14] <http://es.wikipedia.org/wiki/AJAX>

- [15] <http://es.wikipedia.org/wiki/JQuery>
- [16] <http://es.wikipedia.org/wiki/MySQL>
- [17] http://www.apgtic.es/ASI/RAL/RAL-Practica__16.pdf
- [18] http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
- [19] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>
- [20] <http://es.slideshare.net/aimerodriguezrodriguez/que-es-wamp-server>
- [21] <http://es.wikipedia.org/wiki/PhpMyAdmin>
- [22] http://es.wikipedia.org/wiki/Adobe_Dreamweaver
- [23] http://es.wikipedia.org/wiki/MySQL_Workbench
- [24] <http://ganttt-project.uptodown.com/>
- [25] http://es.wikipedia.org/wiki/Microsoft_Word
- [26] http://es.wikipedia.org/wiki/Adobe_Photoshop
- [27] <http://www.visual-paradigm.com/>
- [28] <http://www.programacion.com/php/>
- [29] <http://www.mysql.com/>
- [30] www.php.net/
- [31] <http://es.slideshare.net/Sofylutqm/el-proceso-unificado-3943047>
- [32] <http://es.slideshare.net/JosepSalvadorSotoObregon/unidad-2-planificacion-y-modelado>

